DEPARTMENT OF CHEMICAL ENGINEERING

UNIVERSITY OF CAMBRIDGE

FINITE ELEMENT METHODS FOR THE SOLUTION OF POPULATION BALANCE EQUATIONS

M. Nicmanis

Girton College

A dissertation submitted for the degree of Doctor of Philosophy in

Chemical Engineering at the University of Cambridge

January 1998

Summary

The behaviour of many particulate systems may be described in terms of population balance equations (PBEs). These equations typically relate process conditions (such as crystallizer mean residence times or supersaturation) to number densities of particles with certain properties. In cases where aggregation, breakage and growth of particles is occurring the governing PBEs contain both differential and nonlinear, convoluted integral terms which make them analytically intractable in all but the most ideal cases. Progress in modelling particulate systems is currently being impeded by the lack of reliable, accurate methods for solving these equations.

The work contained within this thesis directly addresses this bottleneck. Finite element methods (FEMs) are developed to solve both steady state and dynamic forms of the PBE. When compared to existing methods the FEMs are found in all cases to be more robust, efficient and capable of achieving accuracies that were previously unattainable.

Much of the success of the above mentioned methods stems from the realization that domains must be carefully specified if high accuracy is to be achieved at reasonable cost. Chapter 4 is entirely dedicated to optimal discretization of the domain. This work represents the first rigorous treatment of this often acknowledged but seldom addressed issue. Numerical simulations demonstrate that such rigorous consideration can improve the computational efficiency and accuracy of a method by orders of magnitude !

Summary

The behaviour of many particulate systems may be described in terms of population balance equations (PBEs). These equations typically relate process conditions (such as crystallizer mean residence times or supersaturation) to number densities of particles with certain properties. In cases where aggregation, breakage and growth of particles is occurring the governing PBEs contain both differential and nonlinear, convoluted integral terms which make them analytically intractable in all but the most ideal cases. Progress in modelling particulate systems is currently being impeded by the lack of reliable, accurate methods for solving these equations.

The work contained within this thesis directly addresses this bottleneck. Finite element methods (FEMs) are developed to solve both steady state and dynamic forms of the PBE. When compared to existing methods the FEMs are found in all cases to be more robust, efficient and capable of achieving accuracies that were previously unattainable.

Much of the success of the above mentioned methods stems from the realization that domains must be carefully specified if high accuracy is to be achieved at reasonable cost. Chapter 4 is entirely dedicated to optimal discretization of the domain. This work represents the first rigorous treatment of this often acknowledged but seldom addressed issue. Numerical simulations demonstrate that such rigorous consideration can improve the computational efficiency and accuracy of a method by orders of magnitude !

A significant breakthrough is made in chapter 3 where an error estimate is derived for the steady state PBE. This estimate bounds the error in a numerical solution from above and ~~and~~ represents the first instance where convergence of a numerical method to the analytical solution has been established for the steady state PBE. The practical implication of this development is that it gives the user the capability to assess quantitatively the quality of an obtained numerical solution without any knowledge of the analytical solution. When combined with domain considerations the error estimate may be used to automate fully the domain discretization procedure. A user needs only to specify a maximum permissible error tolerance and the automatic discretization procedure will find a solution to the problem such that this error tolerance will not be exceeded within any region of the domain.

Acknowledgements

First and foremost I would like to thank my supervisor Dr Mike Hounslow for the excellent supervision, open-minded assessment of new ideas and many piercing insights that he has provided throughout this project.

Many thanks are due to Drs Bill Paterson and Vassilios Vassiliadis for their many helpful suggestions on the numerical aspects of my work. I would also like to express my gratitude to Dr Allan Bramley for his meticulous proof reading of this document and to the past and present members of the crystal group (Jason Barrick, Alan Collier, John Gooch, Hassan Mumtaz, Judith Pearson, Yong Qi, Alison Scott and Ed Wynn) for their constant companionship and many constructive comments on my presentations.

Financial assistance from the ORS committee, Cambridge Commonwealth Trust, Department of Chemical Engineering and Girton College Cambridge has been much appreciated.

Declaration

This thesis is the result of my own independent work, except where otherwise acknowledged. This work was carried out between January 1995 and January 1998 in the Department of Chemical Engineering at the University of Cambridge. Neither this thesis nor any part thereof has been previously submitted or is being concurrently considered for any degree, diploma or other qualification at this or any other institution.

This thesis consists of less than 39500 words.

Publications from this Thesis

Publications

Nicmanis, M., and Hounslow, M.J., (1996), A finite element analysis of the steady state population balance equation for particulate systems : aggregation and growth, *Computers chem. Engng*, **20**, Suppl., pp. S261-S266.

Nicmanis, M., and Hounslow, M.J., (1998), A finite element method for the steady state population balance equation, *A.I.Ch.E. Journal*, Accepted for publication.

Nicmanis, M., and Hounslow, M.J., (1998), An *a Posteriori* error estimate for the steady state population balance equation, *Submitted to International Journal of Numerical Methods in Engineering.*

Nicmanis, M., and Hounslow, M.J., (1998), Automatic refinement for the steady state population balance equation, *Submitted to Computers chem. Engng.*

Nicmanis, M., and Hounslow, M.J., (1998), A finite element method for the dynamic population balance equation, *to be submitted to Computers chem. Engng.*

Conference Presentations

Nicmanis, M., and Hounslow, M.J., (1996), A finite element analysis of the steady state population balance equation for particulate systems : aggregation and growth. Presented at the Sixth European Symposium on Computer Aided Process Engineering, Rhodes, Greece, May 1996.

Nicmanis, M., and Hounslow, M.J., (1997), A computational tool for the design and control of particulate processes. Presented at Design and Control of Particulate Processes IV, Delft, The Netherlands, July 1997.

Contents

Chapter 1

Introduction

In this chapter the fundamental definitions of particulate systems and the mechanisms occurring within such systems are introduced. The relevance and objectives of modelling such systems are discussed from a chemical engineering perspective and a common methodology for modelling processes is outlined. A review of the mathematical formulation of the problem and methods already proposed for solving these formulations is presented. This review is followed by a statement of the objectives of the work contained within this thesis and a justification of the methods used to achieve the stated objectives. The chapter concludes with a statement of why this work is of current relevance to the modelling of particulate systems.

1.1 Definitions : Particulate Systems and Particulate Mechanisms

Particulate systems are those systems consisting of a continuous phase and at least one discrete phase; the individual entities of which will be termed "particles". Associated with each individual particle of the discrete phase will be a number of properties such as : size, colour, shape, composition, age etc. It will be assumed that discrete phases are composed of such large numbers of particles that the distributions (in terms of number or mass) of particles may be observed to vary continuously with respect to each of these properties. For the purposes of this study the manner in which groups of particles are distributed will be described in terms of number density distributions. Readers unfamiliar with this concept should consult the introductory chapters of Randolph and Larson (1988).

Some examples of particulate systems are : a cloud composed of water droplets suspended in air, crystals in a super-saturated solution or even a galaxy composed of stars and planets.

The means by which particle properties change will be termed particulate mechanisms. Of particular interest in this thesis will be the mechanisms by which particles change size, namely :

- *Aggregation* : whereby particles collide, adhere and form new larger particles. Aggregation therefore reduces the total number of particles within a system but does not change the total particulate volume. This term will be used synonymously with agglomeration, coagulation, flocculation and coalescence.
- *Breakage* : whereby a particle is fractured into two or more smaller, separate sub-particles. Breakage therefore increases the total number of particles within the system without changing the total particulate volume. This term will be used synonymously with attrition, comminution, crushing and grinding.

- *Growth* : whereby non-particulate material is deposited on the surface of a particle. Growth therefore increases the total volume of particulate material but does not change the number of particles within the system.

- *Nucleation* : is the process whereby non-particulate matter (of inconsequential volume) condenses to form a particle hence increasing ~~to~~ the total number of particles within the system.

1.2 Relevance of Particulate Systems in Chemical Engineering

Examples of particulate processes may be found throughout the chemical industries. In some cases a product may be isolated by a particulate process, for instance crystallization may be used to manufacture a chemical of high purity, or a mineral slurry may be selectively flocculated. In other cases particulate processes are used to give particle populations more useful physical and chemical characteristics. Examples of this include fine pulverisation of coal for combustion in boilers, granulation of fertilizers and detergents, pelletisation of iron ore to increase its suitability as blast furnace feed and grinding of crystalline pharmaceutical products prior to tabletting. While the study of aerosol dynamics has many important implications in the fields of air pollution and combustion.

1.3 Objective of Modelling Particulate Systems

The ultimate objective of scientists or engineers modelling particulate systems is to develop the capability to predict the distributions of particles with respect to properties of interest for any given set of process variables. Once this objective has

been achieved the model can be used to design particulate processes that are capable of meeting the product specifications demanded by the consumer.

Another industrial motivation for developing computer models is that they give engineers the capability to perform "what if ?" analysis on existing processes. These processes may then be controlled and optimized hence permitting more efficient and cost effective manufacture of the final product.

1.4 Modelling a Process

Initially an "understanding" of the system is required so that a model may be formulated. By "understanding" we imply the possession of a set of assumptions or principles which can reasonably be expected to hold true. For instance it will be shown in the next section that an understanding of the mechanisms occurring within an industrial crystallizer permits continuity/balance principles to be used to derive an equation governing its behaviour.

Such governing equations are frequently derived in terms of parameterizing functions : for instance in the case of an industrial crystallizer growth functions, breakage functions, aggregation and breakage kernels are used to parameterize the equation. If the model is to reflect realistically the behaviour of a physical system then realistic parameterizing functions must somehow be deduced.

The model equations must then be solved. Equations governing particulate systems are typically non-linear and contain both differential and integral terms, hence analytical solutions are scarce in this field. The solution process is usually a non-trivial exercise in numerical methods. If the scientist is to be given any hope of predicting the behaviour of the particulate system then the solution method must be both accurate and robust.

been achieved the model can be used to design particulate processes that are capable of meeting the product specifications demanded by the consumer.

Another industrial motivation for developing computer models is that they give engineers the capability to perform "what if ?" analysis on existing processes. These processes may then be controlled and optimized hence permitting more efficient and cost effective manufacture of the final product.

1.4 Modelling a Process

Initially an "understanding" of the system is required so that a model may be formulated. By "understanding" we imply the possession of a set of assumptions or principles which can reasonably be expected to hold true. For instance it will be shown in the next section that an understanding of the mechanisms occurring within an industrial crystallizer permits continuity/balance principles to be used to derive an equation governing its behaviour.

Such governing equations are frequently derived in terms of parameterizing functions : for instance in the case of an industrial crystallizer growth functions, breakage functions, aggregation and breakage kernels are used to parameterize the equation. If the model is to reflect realistically the behaviour of a physical system then realistic parameterizing functions must somehow be deduced.

The model equations must then be solved. Equations governing particulate systems are typically non-linear and contain both differential and integral terms, hence analytical solutions are scarce in this field. The solution process is usually a non-trivial exercise in numerical methods. If the scientist is to be given any hope of predicting the behaviour of the particulate system then the solution method must be both accurate and robust.

Finally, comparisons should be made between the solutions obtained from the model and those from the physical system under study. If acceptable agreement is achieved it may be assumed that the model has been derived in terms of acceptable assumptions and that the scientist has succeeded in capturing the essential behaviour of the physical system. The model may then be used by other engineers and scientists for the purposes of designing, controlling and optimising relevant systems. In many cases however, unacceptable discrepancies exist between the model predictions and the actual behaviour of a physical system. In these cases the scientist must embark upon the iterative procedure of scrutinizing and modifying the underlying assumptions of the model, resolving the problem and re-assessing the predictions of the modified model.

The steps involved in modelling a physical process may be summarized as follows :

1. Introduce assumptions and adopt principles that will permit a model to be derived.
2. Parameterize the model.
3. Solve the resulting equations.
4. Compare the output of the model to the physical system under study.
5. If acceptable agreement is achieved the modelling process is complete.
6. Otherwise more realistic assumptions, principles and parameterizing functions must be introduced in steps (1) and (2) and the process repeated.

1.5 Mathematical Formulation of the Problem : Population Balance Equations (PBEs)

Drake (1972) attributes the first mathematical description of a particulate process to Smoluchowski (1916 and 1917) who by considering the coagulation problem as a diffusion process, derived an infinite set of governing nonlinear differential equations. Later Müller (1928) used Riemann sums to re-derive these equations as a single nonlinear integro-differential equation. Along similar lines Melzak (1957) derived a nonlinear integro-differential equation describing pure coalescence and particle breakup. All of these researchers assumed coagulation and breakage rates to be functions of particle size only.

A more general framework was given to the problem by Hulburt and Katz (1964) who defined a phase space consisting of internal ($\boldsymbol{r}$) and external coordinates ($\boldsymbol{x}$) and then used continuity/balance considerations to relate the rate of change of the number density distribution ($(n(\boldsymbol{x}, \boldsymbol{r}, t))$ to flux terms and a collection term ($h(\boldsymbol{x}, \boldsymbol{r}, t)$) over this phase space :

$$\begin{aligned} \frac{\partial n(\boldsymbol{x}, \boldsymbol{r}, t)}{\partial t} &+ \sum_{i=1}^{3} \frac{\partial [v_i(\boldsymbol{x}, t) n(\boldsymbol{x}, \boldsymbol{r}, t)]}{\partial x_i} \\ &+ \sum_{j} \frac{\partial [G_j(\boldsymbol{x}, \boldsymbol{r}, t) n(\boldsymbol{x}, \boldsymbol{r}, t)]}{\partial r_j} \\ &= h(\boldsymbol{x}, \boldsymbol{r}, t) \end{aligned} \tag{1.1}$$

In the above equation the external coordinates describe the position of a particle in 3-space at time t while the internal coordinates describe its state; for instance its dimensions, its composition or the number of sub-particles composing it. The flux terms are the changes in the density distribution in a region of the phase space due to the velocities of the particles along the external and internal coordinates respectively. The collection term takes into consideration all other possible sources of change in the density distribution for instance : aggregation and breakage of particles. Nucleation

and particle feed and take-off mechanisms may also be incorporated into the collection term but are more conveniently dealt with as boundary conditions.

If we assume the system is well mixed and restrict consideration to a single internal coordinate; particle volume (v), then the following simplified version of equation (1.1) is obtained :

$$\frac{\partial n(v,t)}{\partial t} + \frac{\partial\big(G(v,t)n(v,t)\big)}{\partial v} = h(v,t) \tag{1.2}$$

If it is assumed that (apart from nucleation) aggregation, growth and breakage are the only means by which particles may change their volume then $G(v,t)$ may be readily identified as the growth rate of a particle of volume v at time t and $h(v,t)$ may be interpreted as the nett collection rate due to aggregation and breakage of particles.

Under the assumptions that only binary collisions occur and that collision frequency is a function of the colliding particles' respective sizes only, Hulburt and Katz (1964) derived expressions for the birth and death rates of particles due to aggregation ($b^a(v,t)$ and $d^a(v,t)$) :

$$b^a(v,t) = \frac{1}{2}\int_0^v \beta(v-w,w,t)n(v-w,t)n(w,t)dw \tag{1.3}$$

and

$$d^a(v,t) = n(v,t)\int_0^\infty \beta(v,w,t)n(w,t)dw \tag{1.4}$$

where the function $\beta(v,w,t)$ is known as the aggregation kernel and is a measure of the frequency with which particles of sizes v and w collide, adhere and form stable aggregates at time t.

Similar considerations have been used by : Bass (1954), Filipov (1961), Gaudin and Meloy (1962) and Gardner and Austin (1962) to derive birth rates and death rates due to breakage ($b^b(v,t)$ and $d^b(v,t)$) which are re-written here as number based quantities :

$$b^b(v,t) = \int_v^\infty \rho(v,w,t)S(w,t)n(w,t)dw \tag{1.5}$$

and

$$d^b(v,t) = S(v,t)n(v,t) \tag{1.6}$$

In the above expressions the breakage function ($\rho(v,w,t)$) is defined so that at time t the probability that a fragment of a particle originally of size w will be broken into the size interval $(v, v+dv)$ is ρdv and the specific rate of breakage ($S(v,t)$) is the rate at which particles of size v break at time t.

The expressions (1.2–1.6) may be combined into a single equation governing the behaviour of particulate systems in which aggregation, breakage and growth (and nucleation if the appropriate boundary condition is imposed) of particles is occurring.

$$\begin{aligned}\frac{\partial n(v,t)}{\partial t} + \frac{\partial\big(G(v,t)n(v,t)\big)}{\partial v} = &\frac{1}{2}\int_0^v \beta(v-w,w,t)n(v-w,t)n(w,t)dw \\ &- n(v,t)\int_0^\infty \beta(v,w,t)n(w,t)dw \\ &+ \int_v^\infty \rho(v,w,t)S(w,t)n(w,t)dw \\ &- S(v,t)n(v,t)\end{aligned} \tag{1.7}$$

For the remainder of this thesis equation (1.7) will be termed the dynamic population balance equation (PBE).

This approach has been further popularised by Randolph and Larson (1988) and can be used to derive a steady state analogue of equation (1.7) :

$$\begin{aligned}\frac{n(v) - n_{in}(v)}{\tau} + \frac{d\big(G(v)n(v)\big)}{dv} = &\frac{1}{2}\int_0^v \beta(v-w,w)n(v-w)n(w)dw \\ &- n(v)\int_0^\infty \beta(v,w)n(w)dw \\ &+ \int_v^\infty \rho(v,w)S(w)n(w)dw \\ &- S(v)n(v)\end{aligned} \tag{1.8}$$

In the above equation $n_{in}(v)$ and $n(v)$ are the number density distributions of the feed and product streams respectively and τ is the mean residence time of the crystallizer. For the remainder of this thesis equation (1.8) will be referred to as the steady state population balance equation.

1.6 Objectives of this Study

The complete procedure for modelling particulate processes as outlined in section 1.4 is beyond the scope of this body of work. Instead the objective has been restricted to developing reliable, robust methods of accurately solving equations (1.7) and (1.8) for the density distribution and moments (moments will be defined in equation ~~(2.1)~~ (2.1) of chapter 2) of these distributions for the full range of particulate mechanisms (aggregation, breakage, growth and nucleation). This work is intended to represent a systematic advancement towards the objectives stated in section 1.3.

The investigation begins by assuming that equations of the forms (1.8) and (1.7) are of some use in modelling particulate systems. These two equations have been selected since they are the simplest possible models likely to be capable of predicting particulate behaviour with respect to a property of frequent industrial interest; particle volume.

The next step is to solve these equations. However, due to the combination of differential, integral and nonlinear terms, few analytical solutions are available. As already mentioned, solving these equations usually requires the application of elaborate numerical methods. The development of such methods is the focal point of this entire ~~thesis~~ dissertation.

This task is performed while keeping the overall objectives of modelling particulate processes in mind. Hence the methods developed in this thesis must be capable of solving equations (1.7) and (1.8) for any set of parameterizing functions and initial/feed distributions. As such, they may be later used to parameterize PBEs for existing systems, to discriminate between likely mechanisms of aggregation, breakage etc. and to assess the validity of the assumptions underlying the PBEs themselves.

It is true that some attempts have already been made to parameterize equations of the forms (1.7) and (1.8); this thesis however still gives precedent to developing methods

for solving the equations since the success of a parametrization method is entirely dependent upon a capability to solve the PBEs accurately.

1.7 Literature Review

Rather than attempt to review every endeavour that has been made to model a particulate process, this literature review will be restricted to those articles relevant to the stated objectives of section 1.6. As such, it will focus on the most significant methods which have been used to solve dynamic and steady state PBEs that are functions of particle size only. Special attention will be paid to those articles where the finite element method has been used. Justification for this emphasis will be given is section 1.8.

1.7.1 Finite Difference Methods

One group of methods frequently used for solving partial differential equations (PDEs) are the finite difference methods (FDMs). In these methods difference quotients are constructed at a finite number of points along the domain of the independent variable. This yields a set of equations that may be solved for the density distribution at each quotient point. FDMs have been successfully used by Kim and Tarbell (1990) and Muhr, David, Villermaux and Jezequel (1996) to solve the dynamic PBE for nucleation and growth. Aggregation and breakage problems are difficult to solve using difference methods since there are no straight-forward approaches to constructing difference quotients for integral terms.

1.7.2 Discretized Population Balances

Discretized population balances (DPBs) are essentially difference methods with additional assumptions introduced so that they may also be used to solve aggregation and breakage problems. In these methods the domain of the independent variable is discretized according to a geometrical progression. Within each sub-domain the density distribution is assumed to be constant and rates of change of this distribution are deduced by considering all possible mechanisms by which a particle may move from one interval to another. A closed set of equations is obtained by enforcing conservation of the relevant moments. Batterham, Hall and Barton (1981) used this method to solve the dynamic PBE for aggregation. Their method correctly conserved particle volume but did not correctly predict the change in particle numbers. This method was extended to correctly predict both particle number and volume for the dynamic PBE for aggregation, growth and nucleation by Hounslow, Ryall and Marshall (1988) and for the steady state case of the same PBE by Hounslow (1990). A similar formulation was derived for the same problem but using a more general geometric discretization by Litster, Smit and Hounslow (1995) and this formulation was later improved upon by Wynn (1996). A similar formulation of the DPB was used by Hill and Ng (1995 and 1996) to solve the dynamic PBE for breakage.

The main short-comings of the DPB formulations are that they assume the density distribution to be constant within each size range and they require the domain to be discretized according to a geometrical progression which is typically sub-optimal (as will be demonstrated in chapter 2). These methods would also be conceptually difficult to generalize to multi-coordinate phase spaces.

1.7.3 Sectional Representations

The first rigorous sectional representation was proposed by Gelbard, Tambour and Seinfeld (1980). In their method the domain was discretized into "sections" and a

property of interest (number, volume, surface area) was assumed constant over each section. Then the rates at which particles were added, removed or remained within a certain section were summed to give the overall rate of change of the property of interest. This method was applied to the dynamic aggregation problem and later extended to model dynamic aggregation, nucleation and growth by Warren and Seinfeld (1985). In later developments Wu and Flagan (1988) derived a sectional representation appropriate to the modelling of the dynamics of small clusters during aerosol formation and growth.

Although these techniques are derived over an arbitrary discretization the accuracy in their predictive capabilities suffers due to the fact that these methods utilize zero order assumptions.

1.7.4 Fixed and Moving Pivot Techniques

In the fixed pivot technique of Kumar and Ramkrishna (1996a) the size domain is discretized into a finite number of intervals and it is assumed that the particle population is concentrated at representative sizes within each sub-domain. Aggregation and breakage rates are deduced for each sub-domain and particles are allocated to the representative sizes so that properties of interest are preserved. This procedure was used to reduce the dynamic PBE to a discrete set of ordinary differential equations in time which were then solved for the evolution of the number of particles within each interval.

The moving pivot technique of Kumar and Ramkrishna (1996b) is an extension of the above method which permits the location of the representative sizes to be moved within each sub-domain. This extension is reported to considerably improve the accuracy of the method.

The main short-coming of these methods is that they are incapable of accurately predicting the density distribution (as will be demonstrated in later chapters by means of a comparison with the finite element method). Inaccuracies in the results obtained are attributed to the discretization procedure. Slightly improved results are obtained by the moving pivot method however the inaccuracies are still considerably larger than those of the finite element method developed later in this thesis.

1.7.5 Method of Weighted Residuals

In the method of weighted residuals (MWR) the solution to a problem is assumed to take the form of a linear combination of basis functions. The equation to be solved is then enforced at discrete points (the collocation method) or by means of weighted integral statements (the Galerkin method). This generates a set of equations that may be solved for the multiplying constants of the linear combination.

The MWR has been used by several researchers to solve dynamic and steady state forms of the PBE. In each case a different basis of functions has been utilized : Singh and Ramkrishna (1975 and 1977) used the MWR with a basis of problem specific polynomials to solve the dynamic aggregation and then the steady state breakage PBE which was later investigated by Hwang and Shih (1982) but this time a basis of block pulse functions was used. Wang and Chang (1983 and 1987) used Legendre functions and later generalized orthogonal polynomials as bases for weighted residual formulations of the steady state PBE for growth. Root-shifted problem-specific polynomials were used by Sampson and Ramkrishna (1984) to solve the dynamic aggregation PBE. The steady state PBE for nucleation, growth and aggregation was investigated by Bhatia and Chakraborty (1996) who scaled the equation prior to solving with polynomial series while Chen, Hwang and Shih (1996) used a basis of wavelet functions to solve the steady state PBE for breakage.

The main problem associated with the MWR is that it is difficult to construct a basis of functions whose linear combination will converge to a function that decreases as rapidly as a typical density distribution. In cases where this can be done the basis is usually problem specific. Hence if the problem was to be re-solved for a different set of feed or initial density distributions then a new basis would need to be constructed.

1.7.6 Finite Element Methods

Like FDMs, finite element methods (FEMs) have established themselves as powerful, all-purpose equation solvers. In these methods the domain is discretized into a finite number of sub-domains or "elements" and the solution is approximated by a linear combination of basis functions (usually polynomials) within each element. Hence the solution takes the form of a piece-wise continuous polynomial over the entire domain. FEMs are better suited to solving PBEs with aggregation and breakage terms than difference schemes since the presence of integral terms does not pose any additional complications; integration of polynomials is a trivial procedure.

As with any discrete method of solving PBEs, correct discretization and truncation of the domain are crucial to its success. In many cases these vital issues have been either completely neglected or poorly addressed thus severely limiting the range of problems that the method is capable of solving.

The first known attempt at using FEMs to solve the PBE was that of Gelbard and Seinfeld (1978). They used a collocation formulation on cubic splines to solve the dynamic PBE for aggregation, growth and nucleation. Their method was derived over a domain that was first truncated and then logarithmically scaled. Although they identified the significance of errors due to domain truncation they offered no suggestions on how appropriate upper limits to the domain should be determined. In their simulations a domain was selected and then held fixed throughout its duration

as was the uniform discretization of their domains. Consequently the accuracy of the predictions of their method deteriorated quite rapidly with time.

Sastry and Gaschignard (1981) formulated a FEM for solving the dynamic PBE for aggregation. The domain was truncated to a finite upper limit, above which the density distribution was assumed to be zero. Once selected this truncation point was held fixed throughout the simulation as was the discretization of the domain. The solution was approximated by a piecewise continuous polynomial the order of which was increased if negative values were attained. Simulations were performed and satisfactory results were achieved but only for very modest extents of aggregation over very small domains. It is however suspected that (as in the previous example) the use of a fixed truncation point and a fixed discretization of the domain would lead to a rapid deterioration in the quality of the results.

Eyre, Wright and Reuter (1988) used cubic B-splines in a collocation formulation to solve the dynamic PBE for aggregation. A singular function was used to map the semi-infinite domain onto a finite interval. An attempt was made to refine the discretization of the domain by equilibrating the arc length within each sub-domain but this approach was found to reduce the accuracy of their method and was subsequently abandoned in later work (Viljoen, Eyre and Wright (1990) and Erasmus, Eyre and Everson (1994)). Predictions of the density distributions by these methods typically incurred relative errors in excess of 10%. The main short-coming of this method is that cubic B-splines perform poorly over non-uniform discretizations hence the possibility of effective mesh refinement is ruled out. Domain mapping was used in an attempt to overcome this complication however this required a mapping parameter (ζ). The correct selection of (ζ) was acknowledged as being "critical to the success of the method" however no suggestions were made as to how appropriate values should be selected.

Steemson and White (1988) proposed a method to solve the steady state PBE for growth and nucleation. In their method the solution was approximated as a piecewise

as was the uniform discretization of their domains. Consequently the accuracy of the predictions of their method deteriorated quite rapidly with time.

Sastry and Gaschignard (1981) formulated a FEM for solving the dynamic PBE for aggregation. The domain was truncated to a finite upper limit, above which the density distribution was assumed to be zero. Once selected this truncation point was held fixed throughout the simulation as was the discretization of the domain. The solution was approximated by a piecewise continuous polynomial the order of which was increased if negative values were attained. Simulations were performed and satisfactory results were achieved but only for very modest extents of aggregation over very small domains. It is however suspected that (as in the previous example) the use of a fixed truncation point and a fixed discretization of the domain would lead to a rapid deterioration in the quality of the results.

Eyre, Wright and Reuter (1988) used cubic B-splines in a collocation formulation to solve the dynamic PBE for aggregation. A singular function was used to map the semi-infinite domain onto a finite interval. An attempt was made to refine the discretization of the domain by equilibrating the arc length within each sub-domain but this approach was found to reduce the accuracy of their method and was subsequently abandoned in later work (Viljoen, Eyre and Wright (1990) and Erasmus, Eyre and Everson (1994)). Predictions of the density distributions by these methods typically incurred relative errors in excess of 10%. The main short-coming of this method is that cubic B-splines perform poorly over non-uniform discretizations hence the possibility of effective mesh refinement is ruled out. Domain mapping was used in an attempt to overcome this complication however this required a mapping parameter (ζ). The correct selection of (ζ) was acknowledged as being "critical to the success of the method" however no suggestions were made as to how appropriate values should be selected.

Steemson and White (1988) proposed a method to solve the steady state PBE for growth and nucleation. In their method the solution was approximated as a piecewise

continuous spline over a uniformally discretized domain. Good accuracy was achieved however this method was never extended to consider aggregation or breakage problems.

Pilinis (1990) proposed a Galerkin FEM to solve the dynamic PBE for aggregation and growth. Linear finite elements were used to span a domain which was logarithmically scaled and then uniformally discretized. Simulations were performed for modest amounts of aggregation and growth and reasonable accuracy was achieved by the method however significant computational time was required.

Tsang and Brock (1983) solved the dynamic PBE for growth. Their method is one of the few that appreciates the importance of correct discretization of the domain. They used a space-time finite element method with a time dependent discretization. Unfortunately this method has not been extended to model aggregation or breakage.

1.8 Which Method Should be Used ?

At the onset of this project a decision had to be made as to which method would be the most suitable for solving the PBEs subject to the stated objectives. The following criteria were used to assess the most likely candidate :

1. The method should be capable of solving the steady state and dynamic PBEs for their density distributions and moments to high accuracy.

2. The method should be capable of modelling the full range of particulate mechanisms : aggregation, breakage, growth and nucleation.

3. The method should be capable of solving PBEs with parameterizing functions (aggregation and breakage kernels, growth and breakage functions, feed/initial density distribution) of any form.

4. The method should be capable of high accuracy predictions for large extents of aggregation, breakage and growth.

5. It should be a straight-forward task to implement domain extrapolation and automatic mesh refinement algorithms into the method.

6. It should be possible to extend the method to higher coordinates systems.

Of all the reviewed methods only the finite element method satisfactorily fulfilled all the above criteria.

Difference methods cannot solve PBEs containing aggregation and breakage terms while the DPB is not amenable to automatic refinement algorithms and would be difficult to generalize to higher coordinate systems. Sectional representations use zero-order approximations that limit the accuracy they can achieve with reasonable computational power and additional assumptions would have to be made in order to implement automatic refinement algorithms. As in the case of the sectional representation the fixed and moving pivot methods would require additional assumptions to regrid from one time step to the next and also suffer from lack of accuracy. The MWRs were excluded since they are unable to match finite element methods in accuracy or ease of implementation.

1.9 Justification for Further Work

Several areas worthy of further work can be identified from the literature review :

- Very little attention has been dedicated to solving the steady state population balance equation. Although not as common in industry as dynamic processes, steady state particulate processes do exist, for example steady state crystallizers. The development of a steady state method would also find application in

parameterizing large scale industrial processes since dynamic simulations would most probably be initialized from a steady state condition. Of the reviewed articles, methods for solving the steady state PBE are proposed using the DPB by Hounslow (1990) and MWRs Singh and Ramkrishna (1977), Hwang and Shih (1982), Wang and Chang (1983), Wang and Chang (1987), Bhatia and Chakraborty (1996) and Chen, Hwang and Shih (1996). While Steemson and White (1988) are the only researchers to propose a finite element method to solve the steady state problem. However this method was limited to solving the trivial case of growth and nucleation. Hence chapter 2 will be dedicated to developing a finite element method for solving the steady state population balance equation for aggregation, breakage, growth and nucleation.

- The steady state PBE is numerically intensive to solve and unlike its dynamic analogue no satisfactory heuristics currently exist that permit pre-specified solution accuracy to be approached. However, recent developments in nonlinear functional analysis now permit a rigorous alternative to these heuristics *ie.* the derivation of an *a posteriori* error estimate. Hence chapter 3 will be dedicated to deriving an error estimate that is capable of quantitatively assessing the "quality" of an obtained numerical solution.

- No systematic methods exist for domain discretization in steady state problems. Current practice is to discretize the domain according to a geometric progression or to use sub-domains of equal size. In many cases refinement according to these schemes can result in very slow convergence to the analytical solution. In chapter 4 an alternative approach to refinement is proposed. This approach is based upon the above mentioned error estimate and permits a maximum acceptable error tolerance to be specified. The new refinement algorithm then automatically finds a discretization such that this error tolerance is not exceeded.

- Although considerably more work has been done on the dynamic PBE, a method capable of solving problems where simultaneous aggregation, breakage, growth

and nucleation are occurring has not yet been proposed. The philosophy taken in this thesis is that all of these phenomena will occur to some extent in any real process. So any method used to model particulate processes should be capable of solving the dynamic PBE with all of these mechanisms occurring. In chapter 5 a finite element method is developed to solve the dynamic PBE for the full range of particulate behaviour.

- Much care must be taken with domain truncation and discretization if a method is to be successful in solving the dynamic PBE. In chapter 5 important domain issues are highlighted and truncation and discretization algorithms are proposed to prevent the problem from becoming stiff or ill-conditioned.

Chapter 2

A Finite Element Method for Solving the Steady State Population Balance Equation

In this chapter a finite element method is derived for solving the steady state population balance equation (PBE). In several numerical case studies this method is shown to be capable of solving the PBE to high accuracy for aggregation, breakage, growth and nucleation problems. Issues of domain truncation, refinement and scaling are discussed and this chapter concludes with a comparison between the derived finite element method and the discretized population balance (DPB) of Litster, Smit and Hounslow (1995).

2.1 Statement of the Problem

As stated in section 1.9 the objective of this chapter is to develop a method capable of solving the steady state population balance equation :

$$\begin{aligned}\frac{n(v)-n_{in}(v)}{\tau}+\frac{d\big(G(v)n(v)\big)}{dv} &= \frac{1}{2}\int_0^v \beta(v-w,w)n(v-w)n(w)dw \\ &\quad - n(v)\int_0^\infty \beta(v,w)n(w)dw \\ &\quad + \int_v^\infty \rho(v,w)S(w)n(w)dw \\ &\quad - S(v)n(v) \qquad \forall\, v\,\epsilon\,(0,\infty]\end{aligned}$$

subject to the boundary condition $n(0) = n_0$ for the number density distribution $n(v)$ and the moments of this distribution. The i^{th} moment of a density distribution is defined as :

$$m_i = \int_0^\infty v^i n(v)dv \tag{2.1}$$

These quantities are of interest due to their physical significance. The zeroth and first moments (m_0 and m_1) with respect to volume, represent the total number and volume of particles in a system while the second moment (m_2) has been shown by Smit, Hounslow and Paterson (1993) to be of use in predicting the onset of gelation.

2.2 Scaled or Unscaled Domain ?

Typical solutions to the PBE tend to be rapidly decreasing functions capable of undergoing changes of several orders of magnitude over the domain of interest, for example an analytical solution to the breakage problem (that will be later studied in section 2.14.2) is shown in figure 2.1. No rigorous methods for optimal discretization of the domain have yet been proposed for PBEs. In an effort to obtain a discretization where the solution changes by similar orders of magnitude over as few sub-domains as possible, several researchers : Gelbard and Seinfeld (1978), Pilinis (1990), Eyre,

Wright and Reuter (1988), Viljoen, Eyre and Wright (1990) and Erasmus, Eyre and Everson (1994), have scaled the domain (according to either logarithmic or singular functions) and then divided it into elements of equal length. In one such example Gelbard and Seinfeld (1978) used the function :

$$x = \frac{\log(v/v_a)}{\log(v_b/v_a)} \tag{2.2}$$

to map their truncated volume domain (v_a, v_b) onto a unit domain $(0, 1)$. The solution over this scaled domain is shown in figure 2.2 where $v_a = 0.0004$ and $v_b = 9.5$. It can be seen that over the domain $(0, 0.8)$ this scaled solution could quite easily be approximated by say four cubic polynomials each of length 0.2. But this corresponds to the region $v \in (0, 1.3)$ in the unscaled domain and only accounts for 91% of the total population by number or 65% by mass. When the unmapped and mapped solutions are plotted against a log y-coordinate (in figures 2.3 and 2.4) it can be seen that the scaled solution becomes much steeper, and hence more difficult to approximate in the region $x \in (0.8, 1)$ than in the unscaled case over the corresponding region $v \in (1.3, 9.5)$. Hence it can be concluded that such scaling functions can make the prediction of the number of small particles easier but at the same time the prediction of the number of large particles becomes more difficult.

Domain scaling also complicates the task of computing the moments of distributions. For example calculation of the second moment requires integration of the function $v^2 n(v)$. When this function is transformed according to the scaling function (2.2) it is multiplied by exponential factors in some cases reducing the integrand to a spike.

For reasons highlighted here, domain scaling will be avoided in this thesis. As an alternative, sub-domains of differing lengths will be used to span the domain. For the purposes of comparison with other methods, sub-domain boundaries will be heuristically located according to a geometric progression in the simulations of section 2.14. A more sophisticated approach to discretization will be introduced later in chapter 4.

Wright and Reuter (1988), Viljoen, Eyre and Wright (1990) and Erasmus, Eyre and Everson (1994), have scaled the domain (according to either logarithmic or singular functions) and then divided it into elements of equal length. In one such example Gelbard and Seinfeld (1978) used the function :

$$x = \frac{\log(v/v_a)}{\log(v_b/v_a)} \tag{2.2}$$

to map their truncated volume domain (v_a, v_b) onto a unit domain $(0, 1)$. The solution over this scaled domain is shown in figure 2.2 where $v_a = 0.0004$ and $v_b = 9.5$. It can be seen that over the domain $(0, 0.8)$ this scaled solution could quite easily be approximated by say four cubic polynomials each of length 0.2. But this corresponds to the region $v \in (0, 1.3)$ in the unscaled domain and only accounts for 91% of the total population by number or 65% by mass. When the unmapped and mapped solutions are plotted against a log y-coordinate (in figures 2.3 and 2.4) it can be seen that the scaled solution becomes much steeper, and hence more difficult to approximate in the region $x \in (0.8, 1)$ than in the unscaled case over the corresponding region $v \in (1.3, 9.5)$. Hence it can be concluded that such scaling functions can make the prediction of the number of small particles easier but at the same time the prediction of the number of large particles becomes more difficult.

Domain scaling also complicates the task of computing the moments of distributions. For example calculation of the second moment requires integration of the function $v^2 n(v)$. When this function is transformed according to the scaling function (2.2) it is multiplied by exponential factors in some cases reducing the integrand to a spike.

For reasons highlighted here, domain scaling will be avoided in this thesis. As an alternative, sub-domains of differing lengths will be used to span the domain. For the purposes of comparison with other methods, sub-domain boundaries will be heuristically located according to a geometric progression in the simulations of section 2.14. A more sophisticated approach to discretization will be introduced later in chapter 4.

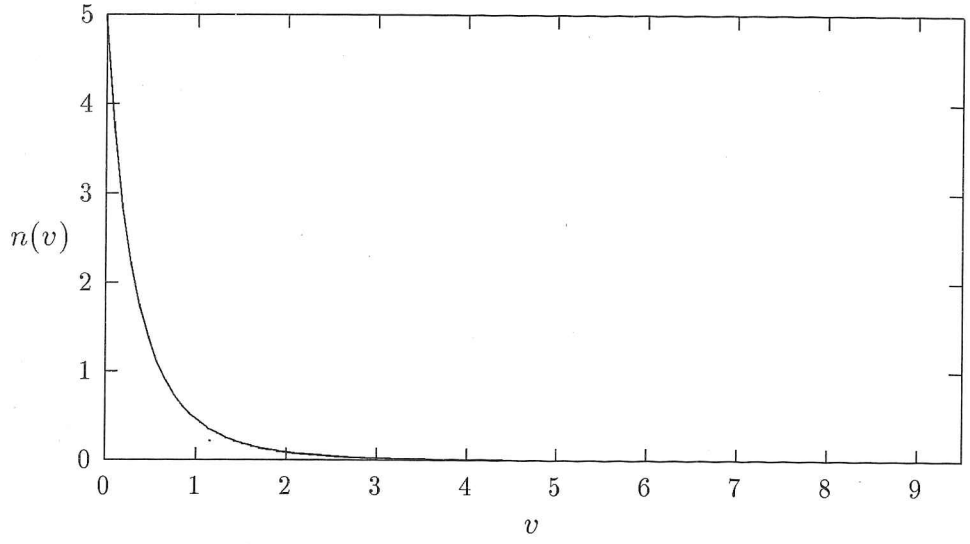

Figure 2.1: An analytical solution to the breakage problem over unscaled volume units

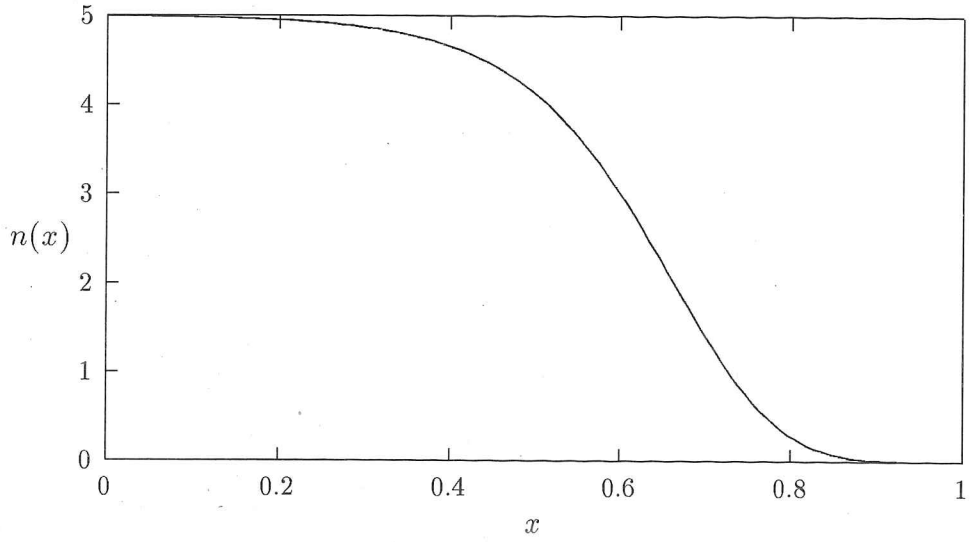

Figure 2.2: The analytical solution to the breakage problem where the domain of the independent variable has been scaled according to the function $x = \ln(v/v_a)/\ln(v_b/v_a)$ where $v_a = 0.0004$ and $v_b = 9.5$.

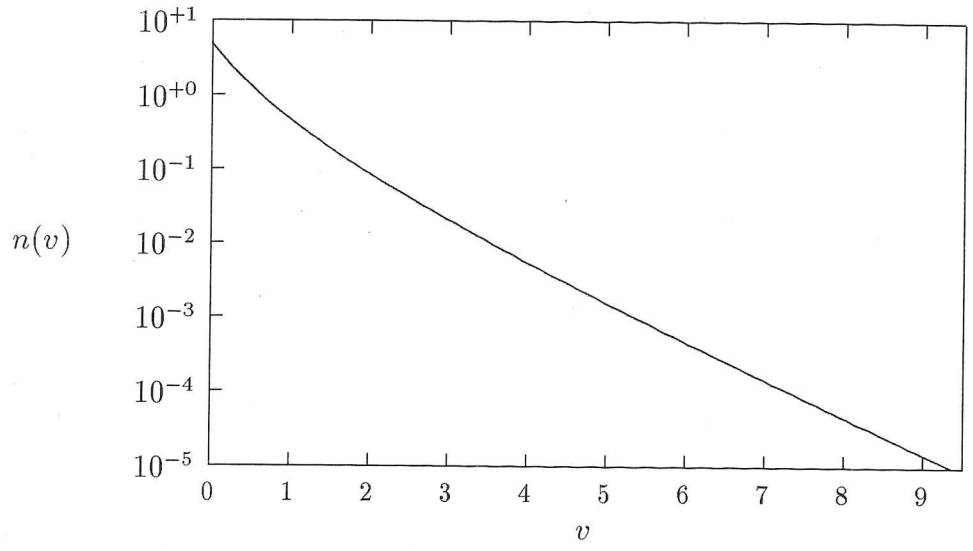

Figure 2.3: The logarithm of analytical solution to the breakage problem over unscaled volume units.

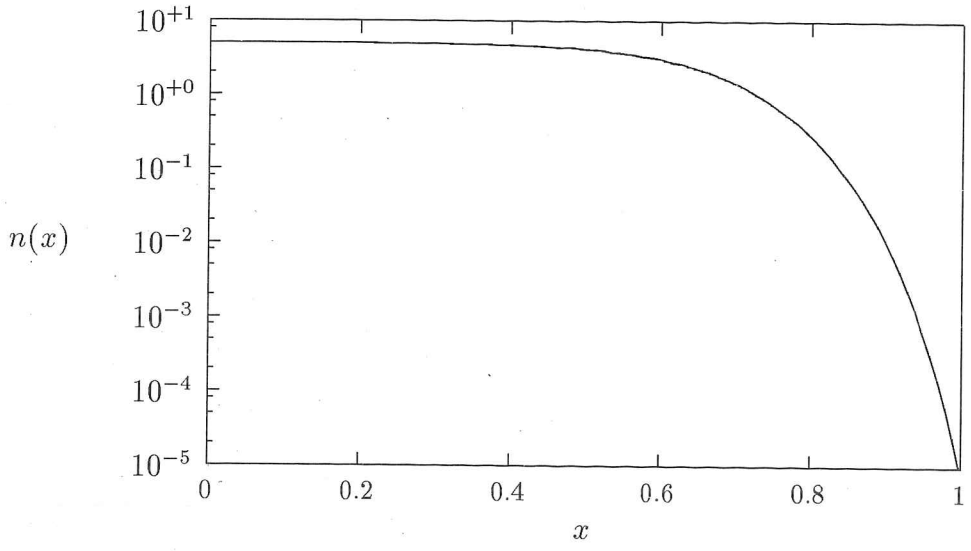

Figure 2.4: The logarithm of the analytical solution to the breakage problem where the domain of the independent variable has been scaled according to the function $x = \ln(v/v_a)/\ln(v_b/v_a)$ where $v_a = 0.0004$ and $v_b = 9.5$.

2.3 Overview of the Finite Element Method

Before embarking upon a full derivation of the FEM for solving the steady state PBE, a brief overview will be given of the essential ideas upon which the finite element method is based.

The fundamental idea behind the finite element method is to divide the domain into a finite number of sub-domains ("elements") and to approximate the solution to the problem as a linear combination of basis functions over each element. The equation to be solved is enforced by means of weighted integral statements which generate the correct number of equations required to solve for the unknown parameters *i.e.* the multiplying constants of the basis functions. Continuity of the solution is imposed at the end-points of each element. This results in a global system of equations which can then be solved for the unknown multiplying constants. Hence, the final solution of the problem takes the form of a piece-wise continuous polynomial function.

These steps may be summarized as follows :

1. Discretize the domain into a finite number of elements.
2. Approximate the solution within each element as a linear combination of basis functions.
3. Formulate a weighted integral statement across each element.
4. Assemble each element into a global system.
5. Solve the system for the unknown multiplying constants.

Each of these steps will be discussed in greater detail in the remaining sections of this chapter.

2.4 Discretization of the Domain

At this stage, notation will be introduced to represent a general discretization of the domain. Methods for determining optimal locations of individual elements will be proposed in chapter 4.

Let $\Omega := (0, v_{max}]$ where $v_{max} < \infty$ is the upper limit of the truncated domain (location of the truncation point will be discussed in section 2.11). Let a general element e be defined as $\Omega_e := (v_a^e, v_b^e]$ then Ω is partitioned into N such elements whereby $v_b^e = v_a^{e-1} \; \forall \; e = 1, \ldots, N$ so that :

$$\cup_{e=1}^{N} \Omega_e = \Omega \qquad \text{and} \qquad \cap_{e=1}^{N} \Omega_e = \emptyset \tag{2.3}$$

2.5 Approximation of the Solution

Consider the e^{th} element $\Omega_e = (v_a^e, v_b^e]$. Within this element the solution to equation (1.8) will be approximated as a linear combination of basis functions :

$$n(v) \approx n_h^e(v) = \sum_{j=1}^{p} n_j^e \psi_j^e(v) \qquad \forall \; v \; \epsilon \; \Omega_e \tag{2.4}$$

In this thesis a basis of cubic Lagrange polynomials will be used :

$$\psi_1^e(v) = \frac{(v - v_2^e)(v - v_3^e)(v - v_4^e)}{(v_1^e - v_2^e)(v_1^e - v_3^e)(v_1^e - v_4^e)}$$

$$\psi_2^e(v) = \frac{(v - v_1^e)(v - v_3^e)(v - v_4^e)}{(v_2^e - v_1^e)(v_2^e - v_3^e)(v_2^e - v_4^e)}$$

$$\psi_3^e(v) = \frac{(v - v_1^e)(v - v_2^e)(v - v_4^e)}{(v_3^e - v_1^e)(v_3^e - v_2^e)(v_3^e - v_4^e)}$$

$$\psi_4^e(v) = \frac{(v - v_1^e)(v - v_2^e)(v - v_3^e)}{(v_4^e - v_1^e)(v_4^e - v_2^e)(v_4^e - v_3^e)}$$

The p points $v_1^e, v_2^e, \ldots, v_p^e$ (in our case $p = 4$) are the zeros of the polynomials and are usually termed the "nodes" of the element. The four above mentioned Lagrange cubics are plotted in figure 2.5. For these functions nodes have been positioned at $v = 0, 1, 2, 3$. The convention $v_1^e = v_a^e$ and $v_p^e = v_b^e$ will be used and it will be assumed

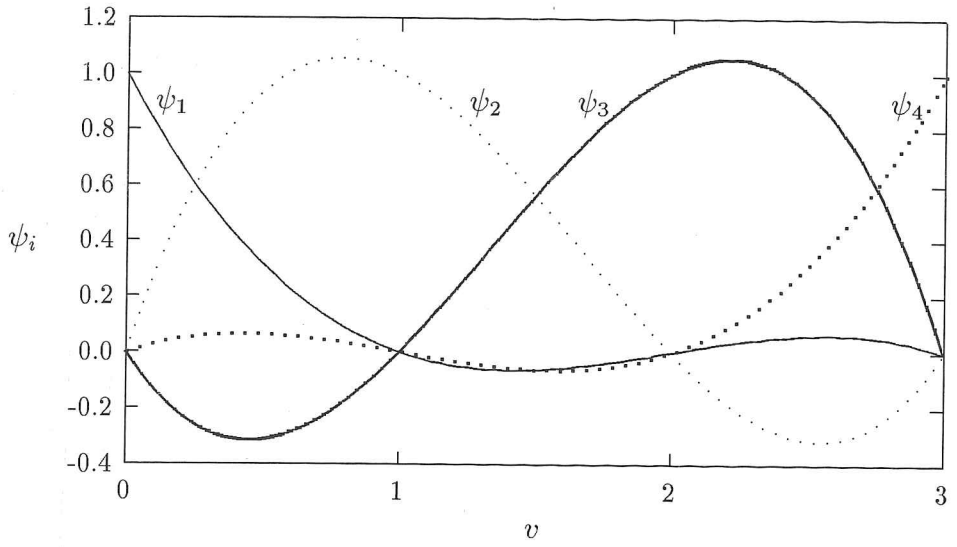

Figure 2.5: Lagrange cubic polynomials with nodes at $v = 0, 1, 2, 3$

that the rest of the nodes are equally spaced within the element. Notice that the Lagrange polynomials have the property :

$$\psi_i^e(v_j^e) = \begin{cases} 0 & \text{if} \quad i \neq j \\ 1 & \text{if} \quad i = j \end{cases} \tag{2.5}$$

Consider approximation (2.4) at node i of element e :

$$n(v_i^e) \approx n_h^e(v_i^e) \;=\; n_1^e\psi_1^e(v_i^e) \;+\; n_2^e\psi_2^e(v_i^e) \;+\cdots+\; n_i^e\psi_i^e(v_i^e) \;+\cdots+\; n_p^e\psi_p^e(v_i^e)$$

By property (2.5) :

$$\begin{aligned} &= \quad 0 \quad + \quad 0 \quad +\cdots+ \quad n_i^e \quad +\cdots+ \quad 0 \\ &= \quad n_i^e \end{aligned}$$

With this choice of basis functions the multiplying constants may be physically interpreted as approximations of the solution at the nodes and hence will be termed "nodal values".

2.6 Formation of a Weighted Residual Expression

Equation (1.8) is reformulated as a weighted residual statement in order to generate a system of equations in terms of solvable parameters (the nodal values of the density distribution).

Equation (1.8) is re-arranged to obtain the following form :

$$n(v)r[v,n] + \tau G(v)\frac{dn(v)}{dv} = n_{in}(v) + \tau b[v,n] \tag{2.6}$$

where three quantities have been introduced to simplify notation :

$$r[v,n] = \left[1 + \tau\left(\bar{d}[v,n] + \frac{dG(v)}{dv}\right)\right] \tag{2.7}$$

$$\bar{d}[v,n] = \frac{d^a(v) + d^b(v)}{n(v)} \tag{2.8}$$

where :

$$d^a(v) = n(v)\int_0^\infty \beta(v,w)n(w)dw \tag{2.9}$$

$$d^b(v) = n(v)S(v) \tag{2.10}$$

$$b[v,n] = b^a(v) + b^b(v) \tag{2.11}$$

where :

$$b^a(v) = \int_0^{v/2} \beta(v-w,w)n(v-w)n(w)dw \tag{2.12}$$

$$b^b(v) = \int_v^\infty \rho(v,w)S(w)n(w)dw \tag{2.13}$$

The arguments of $r, \bar{d}$ and b are used to indicate that they have functional dependencies on both v and the unknown density distribution $n(v)$.

A weighted residual statement is formed from expression (2.6) by multiplying by a weight function $\phi(v)$ and integrating over the domain of element e :

$$\int_{v_a^e}^{v_b^e} \phi(v) \left\{ n(v) r[v,n] + \tau G(v) \frac{dn(v)}{dv} \right\} dv = \int_{v_a^e}^{v_b^e} \phi(v) \left\{ n_{in}(v) + \tau b[v,n] \right\} dv \quad (2.14)$$

At this stage we introduce our approximation of the solution (2.4) :

$$\sum_{j=1}^{p} \int_{v_a^e}^{v_b^e} \phi(v) \left\{ \psi_j^e(v) r[v,\underset{\sim}{n}] + \tau G(v) \frac{d\psi_j^e(v)}{dv} \right\} dv \cdot n_j^e$$

$$= \int_{v_a^e}^{v_b^e} \phi(v) \left\{ n_{in}(v) + \tau b[v,\underset{\sim}{n}] \right\} dv \quad (2.15)$$

The above expression is used to generate a set of p equations by substitution of the set of p weight functions $\{\phi_i^e(v)\}_{i=1}^{p}$ for $\phi(v)$:

$$\sum_{j=1}^{p} \int_{v_a^e}^{v_b^e} \phi_i^e(v) \left\{ \psi_j^e(v) r[v,\underset{\sim}{n}] + \tau G(v) \frac{d\psi_j^e(v)}{dv} \right\} dv \cdot n_j^e$$

$$= \int_{v_a^e}^{v_b^e} \phi_i^e(v) \left\{ n_{in}(v) + \tau b[v,\underset{\sim}{n}] \right\} dv \quad (2.16)$$

where $i = 1, 2, \ldots, p$. Or in matrix notation :

$$A_{ij}^e[\underset{\sim}{\boldsymbol{n}}] n_j^e = F_i^e[\underset{\sim}{\boldsymbol{n}}] \quad (2.17)$$

where :

$$A_{ij}^e = \int_{v_a^e}^{v_b^e} \phi_i^e(v) \left\{ \psi_j^e(v) r[v,\underset{\sim}{n}] + \tau G(v) \frac{d\psi_j^e(v)}{dv} \right\} dv \quad (2.18)$$

and :

$$F_i^e = \int_{v_a^e}^{v_b^e} \phi_i^e(v) \left\{ n_{in}(v) + \tau b[v,\underset{\sim}{n}] \right\} dv \quad (2.19)$$

The argument $(\underset{\sim}{\boldsymbol{n}})$ in equation (2.17) is used to denote that the elements of the A-matrix and the F-vector are dependent upon the vector of unknown nodal values.

2.7 Weight Functions

As stated above the weight functions are introduced into the formulation in order to generate a system of equations that may be solved for the unknown nodal parameters.

The only requirements on these weight functions are that they be i) linearly independent and ii) integrable. Two sets of weight functions giving rise to the collocation and Galerkin formulations are discussed in the next two subsections.

2.7.1 The Collocation Formulation

In the collocation formulation the weight functions are taken to be a set of Dirac delta functions :

$$\phi_i^e(v) = \delta(v - v_i^e) \quad \forall\, i = 1, 2, \ldots\, p \tag{2.20}$$

where the $v_i^e s$ are the collocation points and are selected such that :

$$v_i^e \,\epsilon\, (v_a^e, v_b^e] \quad \forall\, i = 1, 2, \ldots\, p \tag{2.21}$$

With this set of weight functions expressions (2.18) and (2.19) become :

$$A_{ij}^e = \int_{v_a^e}^{v_b^e} \delta(v - v_i^e) \left\{ \psi_j^e(v) r[v, \underset{\sim}{n}] + \tau G(v) \frac{d\psi_j^e(v)}{dv} \right\} dv \tag{2.22}$$

and :

$$F_i^e = \int_{v_a^e}^{v_b^e} \delta(v - v_i^e) \left\{ n_{in}(v) + \tau b[v, \underset{\sim}{n}] \right\} dv \tag{2.23}$$

Further simplification can be made using the definition of the Dirac delta :

$$A_{ij}^e = \psi_j^e(v_i^e) r[v_i^e, \underset{\sim}{n}(v_i^e)] + \tau G(v_i^e) \frac{d\psi_j^e(v_i^e)}{dv} \tag{2.24}$$

$$F_i^e = n_{in}(v_i^e) + \tau b[v_i^e, \underset{\sim}{n}(v_i^e)] \tag{2.25}$$

Hence, in the collocation formulation the PBE is enforced by means of pointwise evaluations within each element.

2.7.2 The Galerkin Formulation

In the Galerkin formulation the basis functions are also used for the weight functions :

$$\phi_i^e(v) = \psi_i^e(v) \quad \forall\ i = 1, 2, \ldots, p \tag{2.26}$$

The expressions (2.18) and (2.19) then become :

$$A_{ij}^e = \int_{v_a^e}^{v_b^e} \psi_i^e(v) \left\{ \psi_j^e(v) r[v,n] + \tau G(v) \frac{d\psi_j^e(v)}{dv} \right\} dv \tag{2.27}$$

and :

$$F_i^e = \int_{v_a^e}^{v_b^e} \psi_i^e(v) \Big\{ n_{in}(v) + \tau b[v,n] \Big\} dv \tag{2.28}$$

Hence in the Galerkin formulation the PBE is enforced by means of integral averages over each element. Since integrals must be evaluated this method is computationally more intensive than the collocation formulation where only point evaluations need to be made. However, it will be seen in the numerical case studies that the Galerkin formulation must be used in some cases to obtain well conditioned systems.

2.8 Nodal Approximations of the Birth and Death Terms

At this stage of the derivation the birth terms for aggregation ($b^a[v,n]$) and breakage ($b^b[v,n]$) and the death term for aggregation ($d^a[v,n]$) are still operators on the unknown density distribution ($n(v)$). These expressions must be reformulated in terms of their nodal values before equation (2.17) can be solved. In this section (and for the remainder of this thesis) it will be assumed that nodes are spaced evenly within each element and that Lagrange cubic polynomials have been used as the interpolation functions. Under these two assumptions a Newton-Cotes fixed point rule can be used to evaluate the integrals. Generalization to polynomials of any order and non-evenly spaced nodes can be made with the combination of an appropriate integration rule and a transformation to an element with evenly spaced nodes.

2.8.1 Approximation of the Death Term for Aggregation

Calculation of $r[v,n]$ in expression (2.7) requires an approximation of the following modified death term for aggregation :

$$\bar{d}^a[v,n] = \frac{d^a[v,n]}{n(v)} = \int_0^\infty \beta(v,w)n(w)dw \tag{2.29}$$

The integrand of expression (2.29) is evaluated at each of the four nodal co-ordinates $(v_1^e, v_2^e, v_3^e, v_4^e)$ of each element. Integration is then performed over each element using a Newton-Cotes $\frac{3}{8}$ rule weighting of these functional evaluations. The modified death term for aggregation results from the sum of these integrations :

$$\bar{d}^a[v,n] \approx \sum_{e=1}^{N}\left(\frac{v_4^e - v_1^e}{3}\left[\frac{3}{8}\beta(v,v_1^e)n_1^e + \frac{9}{8}\beta(v,v_2^e)n_2^e + \frac{9}{8}\beta(v,v_3^e)n_3^e + \frac{3}{8}\beta(v,v_4^e)n_4^e\right]\right) \tag{2.30}$$

in the above approximation N is the number of elements, v_i^e is the volume co-ordinate of the i^{th} node of element e and n_i^e is the i^{th} nodal value of the number density distribution of element e.

2.8.2 Approximation of the Birth Term for Aggregation

To calculate the birth term for aggregation (the steady state form of $b^a(v)$ in equation (1.3)), let $\frac{v}{2}$ lie in element e. The first $e-1$ terms are evaluated as for the death term by making functional evaluations of the integrand of $b^a(v)$ at the nodes of the appropriate elements and applying a Newton-Cotes rule weighting to perform an integration over each element. Hence the contribution from the k^{th} $(k = 1, 2, \ldots, e-1)$ element is :

$$b_k^a(v) = \frac{v_4^k - v_1^k}{3}\left[\frac{3}{8}\beta(v - v_1^k, v_1^k)\bar{n}(v - v_1^k)n_1^k + \frac{9}{8}\beta(v - v_2^k, v_2^k)\bar{n}(v - v_2^k)n_2^k + \frac{9}{8}\beta(v - v_3^k, v_3^k)\bar{n}(v - v_3^k)n_3^k + \frac{3}{8}\beta(v - v_4^k, v_4^k)\bar{n}(v - v_4^k)n_4^k\right] \tag{2.31}$$

in the above expression $\bar{n}(v^*)$ is the cubic polynomial passing through the four points $(v_i^l, n_i^l)_{i=1}^4$ evaluated at v^* where $v^* \epsilon (v_1^l, v_4^l]$. Note that when evaluating $b_k(v)$, v^* does not necessarily lie in element k.

The contribution from the e^{th} element is :

$$b_e^a(v) = h\left[\frac{3}{8}\beta(v - v_1^e, v_1^e)\bar{n}(v - v_1^e)\bar{n}(v_1^e) + \frac{9}{8}\beta(v - \bar{v_2^e}, \bar{v_2^e})\bar{n}(v - \bar{v_2^e})\bar{n}(\bar{v_2^e}) \right.$$
$$\left. + \frac{9}{8}\beta(v - \bar{v_3^e}, \bar{v_3^e})\bar{n}(v - \bar{v_3^e})\bar{n}(\bar{v_3^e}) + \frac{3}{8}\beta(\frac{v}{2}, \frac{v}{2})\{\bar{n}(\frac{v}{2})\}^2\right] \quad (2.32)$$

where : $h = \frac{v/2 - v_1^e}{3}$, $v_2^e = v_1^e + h$ and $v_3^e = v_1^e + 2h$. These contributions are summed to obtain the following approximation of the birth term for aggregation :

$$b^a(v) \approx b_e^a(v) + \sum_{k=1}^{e-1} b_k^a(v) \quad (2.33)$$

2.8.3 Approximation of the Birth Term for Breakage

Let v lie in element e *i.e.* $v \epsilon (v_a^e, v_b^e]$. Then the contribution from the e^{th} element to the birth term for breakage ($b^b[v, n]$ in equation (1.5)) is :

$$b_e^b(v) = h\left[\frac{3}{8}\rho(v, v)S(v)\bar{n}(v) + \frac{9}{8}\rho(v, \bar{v_1^e})S(\bar{v_1^e})\bar{n}(\bar{v_1^e}) + \frac{9}{8}\rho(v, \bar{v_2^e})S(\bar{v_2^e})\bar{n}(\bar{v_2^e}) \right.$$
$$\left. + \frac{3}{8}\rho(v, v_4^e)S(v_4^e)n_4^e\right] \quad (2.34)$$

where : $h = \frac{v_4^e - v}{3}$, v_i^e is the i^{th} node of element e, $\bar{v}_1^e = v + h$, $\bar{v}_2^e = \bar{v}_1^e + h$, n_i^e is the i^{th} nodal value of element e, and $\bar{n}(v)$ is the cubic passing through the four points $(v_i^e, n_i^e)_{i=1}^4$.

The contribution from the k^{th} $(k = e + 1, \ldots, N)$ element is :

$$b_k^b(v) = \frac{v_4^k - v_1^k}{3}\left[\frac{3}{8}\rho(v, v_1^k)S(v_1^k)n_1^k + \frac{9}{8}\rho(v, v_2^k)S(v_2^k)n_2^k + \frac{9}{8}\rho(v, v_3^k)S(v_3^k)n_3^k \right.$$
$$\left. + \frac{3}{8}\rho(v, v_4^k)S(v_4^k)n_4^k \right] \quad (2.35)$$

These contributions are summed to give the following approximation of the birth term for breakage :

$$b^b(v) \approx b^b_e(v) + \sum_{k=e+1}^{N} b^b_k(v) \tag{2.36}$$

2.9 Assembly of the Elements into a Global System

Equation (2.17) results in a system of p equations for each of the N elements of the domain, or a total of $N \times p$ equations. This number is reduced to $N \times (p-1)+1$ (which is the number of unknown nodal values (n_i s) of the global system) by imposing continuity at the end points of the elements. The standard procedure for assembling the elemental matrices is performed as described by Zienkiewicz and Taylor (1989), resulting in a global system of equations. The global matrix consists of a diagonal of N, $p \times p$ blocks : where $i, j = 1, \ldots, N \times (p-1)+1$. The superscript g indicates elements of a global matrix or vector and the argument $\boldsymbol{n}$ has been included to indicate that these elements are dependent upon the unknown nodal values of the density distribution.

2.9.1 Aggregation and Breakage Problems

In cases where aggregation and/or breakage but no growth (*i.e.* $G(v) = 0$) occur, the boundary condition is not required. If both sides of the above equation (2.17) are divided by $r[v, n]$ then the global matrix becomes independent of the vector of unknown nodal values (which reduces the computation expense of solving the global system since the A-matrix must only be decomposed once) :

$$A^g_{ij} n_j = F^g_i[\boldsymbol{n}] \tag{2.37}$$

where the elemental contributions are :

$$A_{ij}^e = \psi_j^e(v_i^e) \tag{2.38}$$

$$F_i^e = \frac{n_{in}(v_i^e) + \tau b[v_i^e, n]}{r[v_i^e, n]} \tag{2.39}$$

for the collocation formulation, or :

$$A_{ij}^e = \int_{v_a^e}^{v_b^e} \psi_i^e(v)\psi_j^e(v)dv \tag{2.40}$$

$$F_i^e = \int_{v_a^e}^{v_b^e} \psi_i^e(v)\frac{n_{in}(v) + \tau b[v, n]}{r[v, n]}dv \tag{2.41}$$

for the Galerkin formulation.

2.9.2 Growth Problems

In all cases where particle growth occurs ($G(v) \neq 0$) the boundary condition must be imposed at $v = 0$ for a unique solution to exist. Hence the number of equations and unknowns is reduced to the following system of $N \times (p-1)$ equations :

$$A_{ij}^g[\boldsymbol{n}]n_j = F_i^g[\boldsymbol{n}] - A_{i1}^g[\boldsymbol{n}]n_0 \tag{2.42}$$

where $i, j = 2, \ldots, N \times (p-1) + 1$. It is possible to re-arrange the above equation so that the A-matrix becomes independent of the vector of unknown density distributions. However, this has been avoided since the resulting system of equations can be difficult to converge.

2.10 Solution of the Global System of Equations

The systems of equations (2.37) and (2.42) are nonlinear and hence iterative schemes are used to solve them.

2.10.1 A Successive Substitution Scheme for Aggregation and Breakage Problems

In aggregation and breakage problems the $s+1^{th}$ estimate of the vector of nodal values is determined from the s^{th} estimate by solving the system :

$$A^g_{ij} n^{s+1}_j = F^g_i[\boldsymbol{n}^s] \tag{2.43}$$

The A^g matrix is decomposed using subroutines described by Press, Teukolsky, Vetterling and Flannery (1992) for banded linear systems. Since this matrix is independent of the nodal values, decomposition need only be performed once. A new improved estimate of the unknown nodal values is obtained by using the back-substitution algorithms (also described by Press, Teukolsky, Vetterling and Flannery (1992)) on the F^g vector which is calculated using the nodal values from the previous iteration.

2.10.2 A Successive Substitution Scheme for Growth Problems

In growth problems the $s+1^{th}$ estimate of the vector of nodal values is determined from the s^{th} estimate by solving the system :

$$A^g_{ij}[\boldsymbol{n}^s] n^{s+1}_j = F^g_i[\boldsymbol{n}^s] - A^g_{i1}[\boldsymbol{n}^s] n_0 \tag{2.44}$$

In this case both the A^g matrix and the F^g vector must be determined using the nodal values from the previous iteration. Decomposition and back-substitution are both performed in each iteration.

In all the numerical case studies investigated in section 2.14 the iterative procedure is initialized using the feed distribution. Iteration is continued until the changes in each of the nodal values are less than 0.01% of ~~those~~ the nodal values of the previous iteration.

2.10.3 Newton-Raphson Methods

Nicmanis (95) attempted to solve the systems of equations (2.37 and 2.42) using the Newton-Raphson methods described by Press, Teukolsky, Vetterling and Flannery (1992) but these methods proved to be difficult, and in many cases computationally expensive, to converge. The latter factor was due to the requirement of this method to assemble and decompose a completely dense $N \times N$ Jacobian matrix for each iteration.

2.11 Domain Truncation

The infinite domain of expression (1.8) must be truncated to a finite upper limit so that it may be spanned by a finite number of elements. This truncation results in an under-estimation of the integrals of expressions (1.4) and (1.5) which reduce equation (1.8) to an approximation of the PBE over a finite domain. It can also be anticipated that the i^{th} moment of the solution will be under-estimated by an amount :

$$m_i^{dte} = \int_{v_{max}}^{\infty} v^i n(v) dv \tag{2.45}$$

where m_i^{dte} is the error incurred in the i^{th} moment due to domain truncation and v_{max} is the upper limit of the finite domain. In most practical applications the density distribution ($n(v)$) asymptotes towards zero at sufficiently large particle volumes so v_{max} can be selected to be sufficiently large that such under-estimation is negligible.

Care must be taken to avoid selection of unnecessarily large values of v_{max} since tail regions of density distributions can be difficult and/or computationally expensive to converge because of the very small values they can attain at large particle volumes.

Little attention has been paid to systematic methods for selecting appropriate upper limits of the domain. Gelbard and Seinfeld (1978) define quantities M_i :

$$M_i = \frac{\int_0^{v_{max}} v^i n(v) dv}{m_i} \tag{2.46}$$

these quantities may be interpreted as the fraction of a moment contained within the truncated domain. They then select v_{max} such that M_0 (the number fraction of particles contained within the truncated domain) and M_1 (the volume fraction of particles contained within the truncated domain) do not differ appreciably from unity. This approach is limited to cases where an analytical expression for m_i is available.

In this section, a systematic method will be suggested for the selection of values of v_{max} that will satisfy the criterion :

$$M_2 > 0.999 \tag{2.47}$$

If $M_2 > 0.999$ then the additional criteria $M_0 > 0.999$ and $M_1 > 0.999$ will also be satisfied. Another motivation for using a criterion based on M_2 (as opposed to M_0 or M_1) is the author's interest in predicting the onset of gelation. This method of domain truncation will be used in all the steady state simulation of chapters 2, 3 and 4. In chapter 5, slight modification to the truncation method will be made so that it may be used in dynamic problems.

2.11.1 Aggregation/Growth Dominant Problems

If the second moment of the solution to a problem (m_2) is greater than the second moment of the feed distribution m_2^{in} :

$$m_2^{in} := \int_0^{\infty} v^2 n_{in}(v) dv \tag{2.48}$$

then the problem will be classified as an aggregation/growth dominant problem. In such problems the value of v_{max} required to satisfy criterion (2.47) can be several

orders of magnitude greater than v_{max}^{in} *i.e.* the upper limit of a domain such that :

$$\frac{\int_0^{v_{max}^{in}} n_{in}(v)dv}{\int_0^{\infty} n_{in}(v)dv} \approx 0.999 \tag{2.49}$$

Selection of an appropriate value of v_{max} is performed as follows :

1. Obtain an order of magnitude estimate of the upper limit (v_{max}^0) : Simulations are performed and plots of the $v^2n(v)$ curve are inspected. Increasing values of the upper limit of the domain are used until the tail region (region where the curve begins to asymptote towards zero for increasing particle sizes) of this curve is located. An order of magnitude estimate is then located on the tail region. It is assumed that v_{max} is slightly under-estimated ie. $M_2 < 0.999$.

2. Refine the partition : The partition is refined (see the next section) to ensure that the finite element solution is a reasonable approximation of the density distribution $n(v)$ over the truncated domain.

3. Extrapolate an exponential function through the last element : An exponential function ($\exp(a + bv)$) is extrapolated through the first and last nodes of the last element. The constants (a and b) are determined such that this exponential passes through the points $(v_a, v_a^2 n(v_a))$ and $(v_b, v_b^2 n(v_b))$, where the subscripts a and b denote the first and last nodes of the last element.

4. M_2 is approximated as follows :

$$M_2 \approx \frac{A_1 + A_2}{A_1 + A_2 + A_3} \tag{2.50}$$

where the quantities A_1, A_2 and A_3 are defined as follows :

$$A_1 := \int_0^{v^0_{max}} v^2 n(v) dv$$

$$\approx \int_0^{v^0_{max}} v^2 n_h(v) dv \tag{2.51}$$

$$A_2 := \int_{v^0_{max}}^{v^f_{max}} v^2 n(v) dv$$

$$\approx \int_{v^0_{max}}^{v^f_{max}} \exp(a + bv) dv$$

$$= \frac{\exp(a + bv^f_{max}) - \exp(a + bv^0_{max})}{b} \tag{2.52}$$

$$A_3 := \int_{v^f_{max}}^{\infty} v^2 n(v) dv$$

$$\approx \int_{v^f_{max}}^{\infty} \exp(a + bv) dv$$

$$= -\frac{\exp(a + bv^f_{max})}{b} \tag{2.53}$$

5. Solve for an improved estimate of v^f_{max} :

   The expression :

$$\frac{A_1 + A_2}{A_1 + A_2 + A_3} = 0.999 \tag{2.54}$$

   may then be solved for v^f_{max} :

$$v^f_{max} = \frac{\ln\left(0.001b\left[\frac{v_b^2 n_h(v_b)}{b} - \int_0^{v^0_{max}} v^2 n_h(v) dv\right]\right) - a}{b} \tag{2.55}$$

which will be an improved estimate of an upper limit that will satisfy criterion (2.47).

2.11.2 Breakage Dominant Problems

In breakage dominant problems the second moment of the solution is smaller than the second moment of the feed distribution :

$$\tau > 0 \Rightarrow m_2 < m_2^{in} \tag{2.56}$$

Two different points, v_{max_1} and v_{max_2} must be located if the solution and its second moment are to be evaluated accurately. The first point v_{max_1} is the volume coordinate such that criterion (2.47) is satisfied. In these problems v_{max}^{in} is a reasonable first estimate of v_{max_1}. An improved estimate of v_{max_1} is obtained using the following procedure :

1. v_{max}^{in} is used as a first estimate of the upper limit to the truncated domain.

2. The quantities M_2^j are defined as follows :

$$M_2^j := \frac{B_1 + B_2 + \ldots + B_j}{\sum_{i=1}^{N} B_i} \tag{2.57}$$

where :

$$B_i := \int_{v_a^i}^{v_b^i} v^2 n_h^i(v) dv \tag{2.58}$$

and : N is the total number of elements spanning the domain, $n_h^i(v)$ is the finite element solution in element i and v_b^i and v_a^i are the upper and lower limits of element i respectively.

3. M_2^js are evaluated for increasing values of j until :

$$M_2^j > 0.999 \tag{2.59}$$

4. Then set :

$$v_{max_1} = v_b^j \tag{2.60}$$

If the birth rate due to breakage is to be calculated to high accuracy at $v = v_{max_1}$ then further extension of the domain is necessary. A second point v_{max_2} is identified such that a second criterion is satisfied :

$$\frac{\int_{v_{max_1}}^{v_{max_2}} \rho(v_{max_1}, w)S(w)n(w)dw}{\int_{v_{max_1}}^{\infty} \rho(v_{max_1}, w)S(w)n(w)dw} > 0.999 \tag{2.61}$$

Using this criterion the point v_{max_2} may be found such that truncation errors in the birth rate due to breakage is negligible for all volume co-ordinates in the domain $v \in (0, v_{max_1})$. The point v_{max_2} is located as follows :

1. An exponential function ($\exp(a + bv)$) is extrapolated through the element containing v_{max_1}. The constants (a and b) are determined such that this function passes through the points (v_a, p_a) and (v_{max_1}, p_b) where :

$$p_a = \rho(v_{max_1}, v_a)S(v_a)n(v_a)$$
$$p_b = \rho(v_{max_1}, v_{max_1})S(v_{max_1})n(v_{max_1})$$

and v_a is the lower limit of the element containing v_{max_1}.

2. The left hand side of criterion (2.61) is approximated as follows :

$$\frac{\int_{v_{max_1}}^{v_{max_2}} \rho(v_{max_1}, w)S(w)n(w)dw}{\int_{v_{max_1}}^{\infty} \rho(v_{max_1}, w)S(w)n(w)dw} \approx \frac{B_1}{B_1 + B_2} \tag{2.62}$$

where :

$$B_1 := \int_{v_{max_1}}^{v_{max_2}} \exp(a + bv)dv$$

$$= \frac{\exp(a + bv_{max_2}) - \exp(a + bv_{max_1})}{b} \tag{2.63}$$

$$B_2 := \int_{v_{max_2}}^{\infty} \exp(a + bv)dv$$

$$= -\frac{\exp(a + bv_{max_2})}{b} \tag{2.64}$$

3. The expression :

$$\frac{B_1}{B_1 + B_2} = 0.999 \tag{2.65}$$

may then be solved for v_{max_2} :

$$v_{max_2} = \frac{\ln(1.999p_b) - a}{b} \tag{2.66}$$

2.12 Heuristic Discretization of the Domain

As an alternative to scaling the domain and then sub-dividing it into elements of equal lengths, domains may also be spanned by elements of increasing length.

Satisfactory results have been achieved by Hounslow (1990) using elements with lengths that increase according to geometrical progressions. In this chapter upper limits of sub-domains will be located according to the following geometrical progression :

$$v_b^e = v_b^1 \left(\frac{v_{max}}{v_b^1}\right)^{\frac{e-1}{N-1}} \tag{2.67}$$

where : superscripts 1 and e denote the first and e^{th} elements, v_b^e denotes the upper limit of element e and N is the total number of elements used to span the domain. The resulting partition is finer at small volumes (where the solution is changing rapidly) and coarser at large volumes (where the solution is changing comparatively slowly). There are clearly many cases where such a discretization will be sub-optimal for instance in problems where the feed distribution is bi-modal. Nevertheless reasonably accurate results can still be achieved and a quite simple method of refinement may be used in conjunction with this discretization.

2.13 Refinement of the Discretization

Once the upper limit of the domain (v_{max}) is established, two adjustable parameters define the geometrical progression 2.67 : i) the length of the first element and ii) the number of elements in the partition. The values of these parameters are selected using the following procedure :

1. The domain is truncated, partitioned and the problem is solved for the density distribution and its moments.
2. The length of the first element is reduced and the problem re-solved until further reductions result in insignificant changes in the 0^{th} moment.
3. The number of elements is increased and the problem re-solved until further increases in number of elements result in insignificant changes to the value of the 2^{nd} moment.

The main short-coming of the refinement scheme suggested above is that it requires much guess-work, for instance : how much should the length of first element be reduced before the problem is re-solved ? Incorrect guesses can significantly add to the total computational cost required to obtain a satisfactory answer. A more rigorous

approach that entirely eliminates the need for such guess work will be proposed in chapter 4.

2.14 Numerical Case Studies

The finite element method was applied to four cases of the PBE for which analytical solutions are available. Aggregation, breakage and growth were each studied as isolated phenomena in the first three cases while a problem of combined aggregation, growth and nucleation was investigated in the final case.

2.14.1 An Aggregation Problem

If the breakage function, the specific rate of breakage and the growth function are all set to zero :

$$\rho(v,w) = 0 \quad \forall \quad v,w \tag{2.68}$$

$$S(v) = 0 \quad \forall \quad v \tag{2.69}$$

$$G(v) = 0 \quad \forall \quad v \tag{2.70}$$

then the steady state PBE for aggregation is obtained :

$$\frac{n(v) - n_{in}(v)}{\tau} = \int_0^{v/2} \beta(v-w,w)n(v-w)n(w)dw - \int_0^{\infty} \beta(v,w)n(w)dw \tag{2.71}$$

For the idealized case of an exponential feed distribution :

$$n_{in}(v) = \frac{N_0}{v_0} \exp\left(\frac{-v}{v_0}\right) \tag{2.72}$$

and a size independent aggregation kernel :

$$\beta(v,w) = \beta_0 \quad \forall \quad v,w \tag{2.73}$$

the following analytical solution has been derived by Hounslow (1990) :

$$n(v) = \frac{N_0}{v_0} \left(\frac{I_0 \left[\frac{-tv}{v_0(1+2t)} \right] + I_1 \left[\frac{-tv}{v_0(1+2t)} \right]}{\sqrt{1+2t} \exp \left[\frac{(1+t)v}{(1+2t)v_0} \right]} \right) \tag{2.74}$$

where : I_0 and I_1 are modified Bessel functions of the first kind, of zeroth and first orders and $t = \beta_0 N_0 \tau$. An asymptotic expansion of the above solution has been derived by Hounslow (1997) for $v \to \infty$:

$$n(v) = \frac{\exp \left[\frac{-v}{2v_0 t} \right]}{\sqrt{\pi}(2t)^2 \left[\frac{v}{2v_0 t} \right]^{\frac{3}{2}}} \tag{2.75}$$

and is used when :

$$\frac{tv}{v_0(1+2t)} > 700 \tag{2.76}$$

since the values of the modified Bessel functions exceed the numerical range of most processors for such arguments.

Analytical expressions for the zeroth, first and second moments of the density distribution have been derived by Smit, Hounslow and Paterson (1993) to be :

$$m_0 = \frac{-1 + \sqrt{1 + 2\beta_0 m_0^{in} \tau}}{\beta_0 \tau} \tag{2.77}$$

$$m_1 = m_1^{in} \tag{2.78}$$

$$m_2 = \beta_0 m_1^2 \tau + m_2^{in} \tag{2.79}$$

The above expressions have been used to evaluate the analytical values of the moments for each of the simulations performed. These values are listed in Table 2.1. Also tabulated is the index of aggregation (I_{agg}) which has been defined by Hounslow (1990)

as :

$$I_{agg} = 1 - \frac{m_0}{m_0^{in}} \tag{2.80}$$

and is used as a quantitative measure of the extent to which complete aggregation has been achieved. This index can attain values between zero (no aggregation) and one (complete aggregation).

case	τ	m_0	m_1	m_2	I_{agg}
1a	2×10^2	9.51×10^{-2}	1	202	0.9049
1b	2×10^4	9.95×10^{-3}	1	20002	0.9901
1c	2×10^6	9.99×10^{-4}	1	200002	0.9990

Table 2.1: Analytical moments of the density distribution and indices of aggregation for the aggregation problem.

Truncation and mesh refinement were performed as previously described. An initial estimate of the truncation point (v^0_{max}) was located by visual inspection. The extrapolation procedure for aggregation/growth dominant problems was then used to obtain an improved estimate for a truncation point (v_{max}) satisfying criterion (2.47). These estimates are recorded below in Table (2.2). Also tabulated are the quantities M_2^0 (M_2 of equation (2.46) using v^0_{max} as an upper limit) and M_2 (which uses v_{max} as an upper limit). These quantities were evaluated using the analytical expressions (2.74) and (2.75) as a check on the appropriateness of the truncation points. In the final two columns of Table (2.2) the length of the first element and the number of elements used to span the truncated domain are recorded. These are the parameters of the geometrical progression used in the final refined partition to generate the results listed in Table (2.3).

The collocation formulation of the aggregation problem was solved using the successive substitution strategy (2.43). The finite element predictions of the number density function ($n(v)$) are shown in figure (2.6). Large relative errors are most likely

case	v^0_{max}	M^0_2	v_{max}	M_2	v_1	N
1a	2.500×10^3	0.9940	3.306×10^3	0.9990	1.0	32
1b	2.500×10^5	0.9941	3.294×10^5	0.9990	1.5	40
1c	2.500×10^7	0.9941	3.403×10^7	0.9993	2.0	45

Table 2.2: Domain truncation and partition parameters used in the aggregation problem.

to be present in the tail region of the density distribution. For this reason the results are plotted on log-log axes. The solid lines represent the analytical solution while the symbols are the nodal values of the finite element solution for the first and last node of each element. Excellent agreement can be observed between the analytical and numerical values. This agreement will be quantified by the mean square relative error.

A mean square relative error in the number of particles predicted in an interval can be defined as a measure of the quality of a solution :

$$\frac{1}{N}\text{SSRE} = \frac{1}{N}\sum_{i=1}^{N}\left(\frac{m(i)_0^{fea} - m(i)_0^{ana}}{m(i)_0^{ana}}\right)^2 \tag{2.81}$$

In the above expression $m(i)_0$ is the number of particles in the i^{th} partition and the superscripts *fea* and *ana* denote predictions of the finite element solution and the analytical solution respectively. Values of $\frac{1}{N}$SSRE and the percentage errors in the moments of the finite element solution ($E\ m_i$) are recorded in Table (2.3). Also tabulated are the number of equations comprising the system (N_{eq}), the number of iterations required for the system to converge (N_{it}) and CPU requirements of the Sun Ultra Enterprise II that was used for the simulations (CPU). One CPU unit corresponds approximately to one second of real time. In all cases very high quality predictions were made of the density distribution (as indicated by the small $\frac{1}{N}$SSRE values) and its moments (as indicated by small percentage errors). More complete

case	v^0_{max}	M^0_2	v_{max}	M_2	v_1	N
1a	2.500×10^3	0.9940	3.306×10^3	0.9990	1.0	32
1b	2.500×10^5	0.9941	3.294×10^5	0.9990	1.5	40
1c	2.500×10^7	0.9941	3.403×10^7	0.9993	2.0	45

Table 2.2: Domain truncation and partition parameters used in the aggregation problem.

to be present in the tail region of the density distribution. For this reason the results are plotted on log-log axes. The solid lines represent the analytical solution while the symbols are the nodal values of the finite element solution for the first and last node of each element. Excellent agreement can be observed between the analytical and numerical values. This agreement will be quantified by the mean square relative error.

A mean square relative error in the number of particles predicted in an interval can be defined as a measure of the quality of a solution :

$$\frac{1}{N}\text{SSRE} = \frac{1}{N}\sum_{i=1}^{N}\left(\frac{m(i)_0^{fea} - m(i)_0^{ana}}{m(i)_0^{ana}}\right)^2 \tag{2.81}$$

In the above expression $m(i)_0$ is the number of particles in the i^{th} partition and the superscripts *fea* and *ana* denote predictions of the finite element solution and the analytical solution respectively. Values of $\frac{1}{N}$SSRE and the percentage errors in the moments of the finite element solution ($E\ m_i$) are recorded in Table (2.3). Also tabulated are the number of equations comprising the system (N_{eq}), the number of iterations required for the system to converge (N_{it}) and CPU requirements of the Sun Ultra Enterprise II that was used for the simulations (CPU). One CPU unit corresponds approximately to one second of real time. In all cases very high quality predictions were made of the density distribution (as indicated by the small $\frac{1}{N}$SSRE values) and its moments (as indicated by small percentage errors). More complete

discussion of these values will be provided later in section 2.16. It may also be noted that a significant number of iterations (16377) were required by the successive substitution algorithm to solve case 1c of the aggregation problem. Justification will also be given is section 2.16 for the continued use of this apparently inefficient solution algorithm.

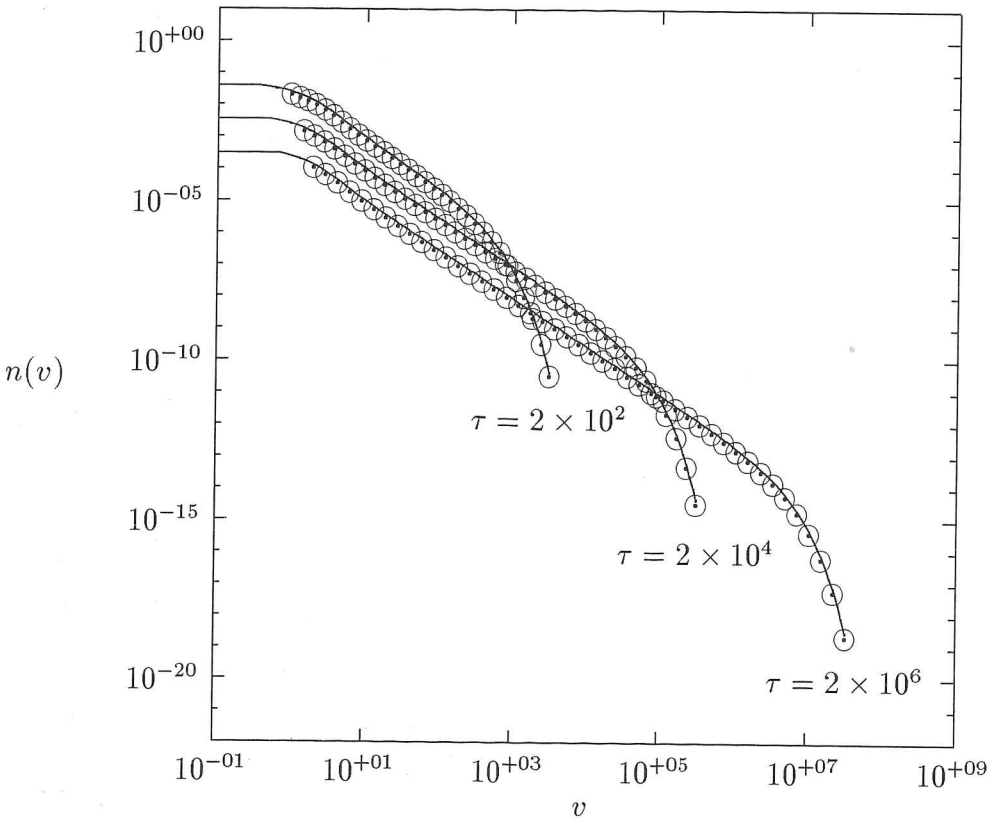

Figure 2.6: FEM predictions of the number density function $(n(v))$ for the aggregation problem where $n_{in}(v) = \frac{N_0}{v_0} \exp(-v/v_0)$, $\beta(v, w) = \beta_0$ and $N_0 = v_0 = \beta_0 = 1$ for the three mean residence times $\tau = 2 \times 10^2, 2 \times 10^4$ and 2×10^6.

case	$E\ m_0$	$E\ m_1$	$E\ m_2$	$\frac{1}{N}SSRE$	N_{eq}	CPU	N_{it}
1a	0.003%	0.023%	0.134%	2.008×10^{-9}	97	$4u$	257
1b	0.016%	0.073%	0.192%	7.256×10^{-8}	121	$64u$	2121
1c	0.051%	0.095%	0.139%	6.325×10^{-7}	136	$573u$	16377

Table 2.3: Errors in moments and mean square relative errors of the finite element solution to the aggregation problem.

2.14.2 A Breakage Problem

If the aggregation kernel and the growth function are both set to zero :

$$\beta(v,w) = 0 \quad \forall\, v,w \tag{2.82}$$

$$G(v) = 0 \quad \forall\, v \tag{2.83}$$

then the PBE for breakage is obtained :

$$\frac{n(v) - n_{in}(v)}{\tau} = \int_v^\infty \rho(v,w)S(w)n(w)dw - S(v)n(v) \tag{2.84}$$

As in the aggregation problem an exponential feed distribution will be used. For breakage with a binary breakage function and size dependent rate of breakage :

$$\rho(v,w) = \frac{2}{w}, \qquad S(v) = v \tag{2.85}$$

the following solution for the density distribution will be derived in the appendix as :

$$n(v) = \frac{N_0[(1+\tau v)^2 + 2\tau v_0(1+\tau[v_0+v])]}{v_0(1+\tau v)^3 \exp\left(\frac{v}{v_0}\right)} \tag{2.86}$$

The constants N_0 and v_0 were set to unity and the breakage problem was solved for the three time constants listed in Table (2.4).

An analytical expression for the zeroth moment can be derived by integrating the PBE for breakage over the volume domain $v\ \epsilon\ (0,\infty)$, which gives :

$$m_0 = \tau m_1 + m_0^{in} \tag{2.87}$$

case	τ	m_0	m_1	m_2	I_{bre}
2a	1×10^{0}	2	1	$1.192 \times 10^{+0}$	0.5000
2b	1×10^{2}	101	1	8.157×10^{-2}	0.9901
2c	1×10^{4}	10001	1	1.727×10^{-3}	0.9999

Table 2.4: Analytical moments of the density distribution and indices of breakage.

It is assumed that total particle volume is conserved for systems where breakage is the only particulate mechanism, hence the first moment of the feed distribution will be the same as that of the product distribution.

$$m_1 = m_1^{in} = \int_0^{\infty} v n_{in}(v) dv \tag{2.88}$$

The above analytical expressions were used to calculate the zeroth and first moments for each of the time constants listed in Table (2.4).

An analytical expression for the second moment can only be derived in terms of the third moment (which can only be derived in terms of the forth moment and so forth), hence semi-analytical values of the second moment were obtained by numerical integration of the function $v^2 n(v)$ where $n(v)$ is the solution to the breakage PBE (2.86).

An index of breakage I_{bre} can be defined by analogy with the index of aggregation :

$$I_{bre} = 1 - \frac{m_0^{in}}{m_0} \tag{2.89}$$

Analytical and semi-analytical evaluation of the moments of the solution and indices of breakage are recorded in Table (2.4).

Domain truncation was performed using the method described previously for breakage dominant problems. Truncation points (initial and final estimates), corresponding M_2 values and the parameters of the refined geometrical progression are recorded in Table (2.5).

case	τ	m_0	m_1	m_2	I_{bre}
2a	1×10^{0}	2	1	$1.192 \times 10^{+0}$	0.5000
2b	1×10^{2}	101	1	8.157×10^{-2}	0.9901
2c	1×10^{4}	10001	1	1.727×10^{-3}	0.9999

Table 2.4: Analytical moments of the density distribution and indices of breakage.

It is assumed that total particle volume is conserved for systems where breakage is the only particulate mechanism, hence the first moment of the feed distribution will be the same as that of the product distribution.

$$m_1 = m_1^{in} = \int_0^{\infty} v n_{in}(v) dv \tag{2.88}$$

The above analytical expressions were used to calculate the zeroth and first moments for each of the time constants listed in Table (2.4).

An analytical expression for the second moment can only be derived in terms of the third moment (which can only be derived in terms of the forth moment and so forth), hence semi-analytical values of the second moment were obtained by numerical integration of the function $v^2 n(v)$ where $n(v)$ is the solution to the breakage PBE (2.86).

An index of breakage I_{bre} can be defined by analogy with the index of aggregation :

$$I_{bre} = 1 - \frac{m_0^{in}}{m_0} \tag{2.89}$$

Analytical and semi-analytical evaluation of the moments of the solution and indices of breakage are recorded in Table (2.4).

Domain truncation was performed using the method described previously for breakage dominant problems. Truncation points (initial and final estimates), corresponding M_2 values and the parameters of the refined geometrical progression are recorded in Table (2.5).

case	v^0_{max}	M^0_2	v_{max_1}	v_{max_2}	M_2	v_1	N
2a	12	1.0000	9.2373	15.629	0.9991	1×10^{-1}	15
2b	12	1.0000	7.4129	13.668	0.9992	1×10^{-3}	30
2c	12	1.0000	6.7777	12.981	0.9993	1×10^{-5}	50

Table 2.5: Domain truncation and partition parameters used in the breakage problem.

As in the previous case the successive substitution scheme (2.43) was used to solve the collocation formulation of the problem. A log-log plot of the finite element predictions of the number density function ($n(v)$) is shown in figure (2.7). The symbols represent the nodal values of the finite element solution for the first and last node of each element. These values may be observed to be in excellent agreement with the analytical solution which is shown as a solid line. It should be noted that the solutions are only plotted in the interval $(0, v_{max_1})$ since the solution over the interval (v_{max_1}, v_{max_2}) contain significant truncation errors and was only computed in order to reduce the truncation error in the birth rate due to breakage within the interval $(0, v_{max_1})$. Moments of the numerical solution were evaluated and the errors in these moments as well as the $\frac{1}{N}SSRE$ values were recorded in Table (2.6). The low values of all these errors is an indication of a very high quality numerical solution. The number of equations solved, CPU requirements of the simulations and the number of iterations required for convergence were also recorded in Table (2.6). In case 2c, convergence of the numerical solution was achieved in significantly fewer iterations (79) than in case 1c eventhough the analytical solution differs from the feed distribution by a similar order of magnitude ($I_{bre} \approx I_{agg} \approx 0.999$).

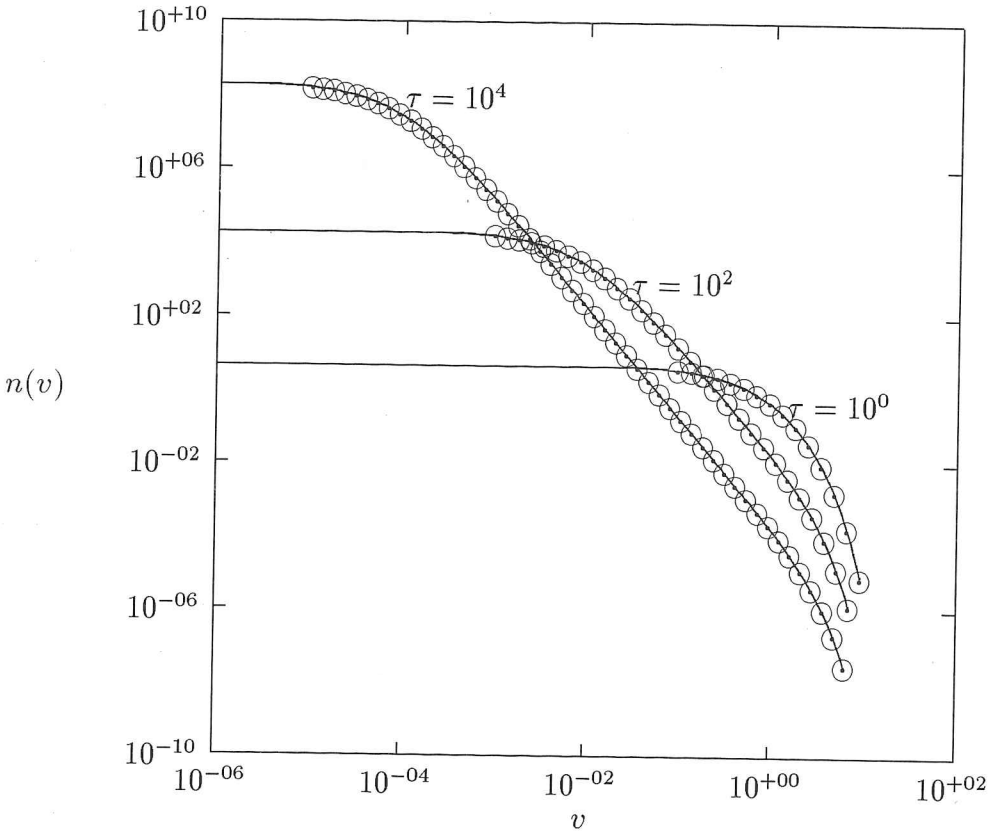

Figure 2.7: FEM predictions of the number density function ($n(v)$) for the breakage problem where $n_{in}(v) = \frac{N_0}{v_0}\exp(-v/v_0)$, $\rho(v,w) = 2/w$, $S(v) = v$ and $N_0 = v_0 = 1$ for the three mean residence times $\tau = 10^0, 10^2$ and 10^4.

case	$E\ m_0$	$E\ m_1$	$E\ m_2$	$\frac{1}{N}SSRE$	N	CPU	N_{it}
2a	0.003%	0.047%	0.142%	1.105×10^{-5}	15	$< 1u$	15
2b	0.192%	0.189%	0.278%	1.160×10^{-4}	30	$< 1u$	36
2c	0.270%	0.267%	0.308%	2.054×10^{-4}	50	$298u$	79

Table 2.6: Errors in moments and mean square relative errors of the finite element solution to the breakage problem.

2.14.3 A Growth Problem

If the aggregation kernel, breakage function and the specific rate of breakage are all set to zero :

$$\beta(v,w) = 0 \quad \forall\, v,w \tag{2.90}$$

$$\rho(v,w) = 0 \quad \forall\, v,w \tag{2.91}$$

$$S(v) = 0 \quad \forall\, v \tag{2.92}$$

then the PBE for growth is obtained :

$$\frac{n(v) - n_{in}(v)}{\tau} + \frac{d}{dv}\left(G(v)n(v)\right) = 0 \tag{2.93}$$

The feed distribution :

$$n_{in}(v) = \frac{N_0}{v_0} v \exp\left(-\frac{v}{v_0}\right) \tag{2.94}$$

was used and since this PBE contains a derivative term the boundary condition $n(0)$ must be specified if a unique solution is to be obtained. This condition was set to zero :

$$n(0) = 0 \tag{2.95}$$

This combination of feed distribution and boundary condition was selected to give a growth problem without nucleation, but with a continuous solution.

A constant growth function was selected for this problem :

$$G(v) = G_0 \tag{2.96}$$

Although not physically realistic, a constant growth rate results in a PBE which may be routinely solved. The following analytical solution to this problem may be obtained using an integrating factor :

$$n(v) = N_0 \frac{\tau G_0 v_0 \exp(-\frac{v}{\tau G_0}) + [v_0 v - \tau G_0(v + v_0)] \exp(-\frac{v}{v_0})}{(\tau G_0 - v_0)^2} \tag{2.97}$$

Simulations were performed with N_0 and v_0 set to unity for each of the three values of τG_0 listed in Table (2.7).

Analytical expressions for the zeroth, first and second moments may be obtained by multiplying the PBE for breakage by v^i (where i is the appropriate index) and integrating over the volume domain. This procedure results in the following three expressions :

$$m_0 = m_0^{in} \tag{2.98}$$

$$m_1 = m_1^{in} + \tau G_0 m_0 \tag{2.99}$$

$$m_2 = m_2^{in} + 2\tau G_0 m_1 \tag{2.100}$$

Evaluations of these moments are presented for each of the values of τG_0 considered in Table (2.7).

case	τG_0	m_0	m_1	m_2
3a	1×10^{-1}	1	2.1	6.42
3b	$1 \times 10^{+1}$	1	12	246
3c	$1 \times 10^{+3}$	1	1002	2004006

Table 2.7: Analytical moments of the density distribution for the growth problem.

case	v_{max}^0	M_2^0	v_{max}	M_2	v_1	N
3a	1×10^0	0.9893	1.369×10^1	0.9993	0.1	20
3b	1×10^2	0.9972	1.140×10^2	0.9991	0.4	25
3c	1×10^4	0.9972	1.144×10^4	0.9991	0.5	30

Table 2.8: Domain truncation and partition parameters used in the growth problem.

Domain truncation was performed as described previously for aggregation/growth dominant problems. Truncation points, corresponding M_2 values and the parameters of the refined geometrical progression are recorded in Table (2.8).

The successive substitution strategy (2.44) was used to solve both collocation and Galerkin formulations of this problem. It was found that the collocation formulation resulted in ill-conditioned global matrices. The magnitudes of the derivatives of the interpolation functions increase very rapidly as the element lengths decrease in size. Partitioning the domain according to the geometric progression (2.67) results in a global matrix with entries in the top left corner that are large enough to cause matrix ill-conditioning. An example where this has occurred is shown in figure (2.8).

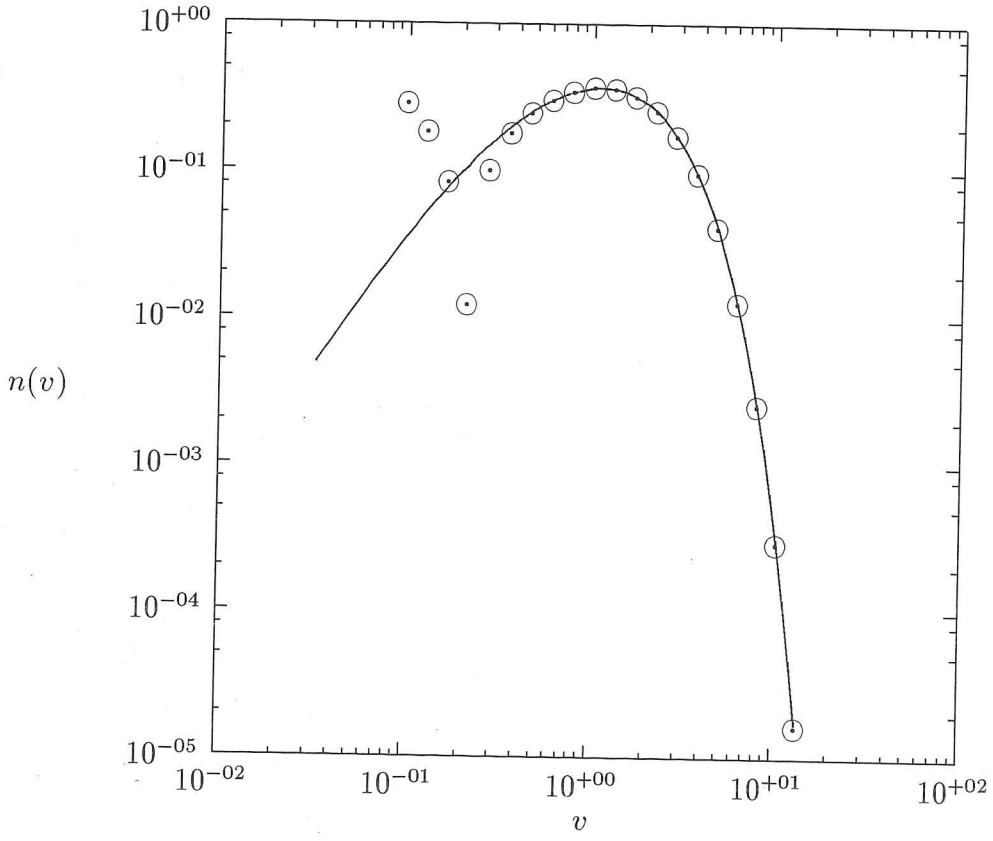

Figure 2.8: The poor solution of the collocation formulation resulting from matrix ill-conditioning in case 3a of the growth problem.

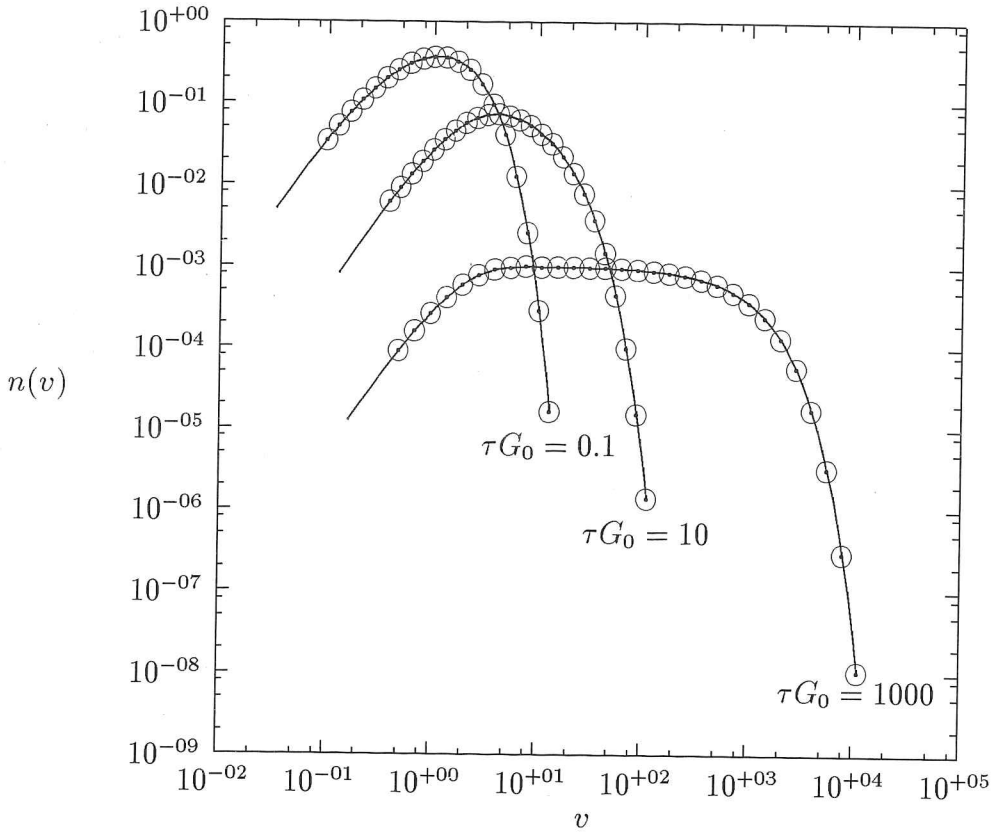

Figure 2.9: More accurate solutions of the collocation-Galerkin formulation of the growth problem where $n_{in}(v) = \frac{N_0 v}{v_0} \exp(-v/v_0)$, $G(v) = G_0$ and $N_0 = v_0 = 1$ for the three cases where $\tau G_0 = 0.1, 10$ and 1000 .

A much better conditioned system is obtained if the Galerkin formulation of the problem is used in all elements where :

$$\text{element length } < 10\tau G_{max}(v) \tag{2.101}$$

and where $G_{max}(v)$ is the maximum value attained by the growth function within the element.

When the collocation-Galerkin formulation was used to solve the growth problem the finite element predictions of the density distribution ($\odot$) may be observed, in figure (2.9), to be in excellent agreement with the analytical solutions (solid lines).

The errors in the moments of the solution, the mean relative error in the number of particles predicted in an interval, the number of equations to be solved, the CPU requirements of the simulations and the number of iterations required for convergence are recorded in Table (2.9). Once again the high quality of the numerical solution is quantified by very small errors in the numerically obtained moments and small $\frac{1}{N}$SSRE values. It should also be noted that only one iteration was required to solve the growth PBE since it is a linear system.

case	$E\ m_0$	$E\ m_1$	$E\ m_2$	$\frac{1}{N}SSRE$	N	CPU	N_{it}
3a	0.001%	0.013%	0.064%	1.527×10^{-5}	10	$< 1u$	1
3b	0.006%	0.008%	0.080%	5.241×10^{-6}	15	$< 1u$	1
3c	0.015%	0.003%	0.067%	8.688×10^{-9}	20	$< 1u$	1

Table 2.9: Errors in moments and mean square relative errors of the finite element solution to the growth problem.

2.14.4 A Problem of Combined Aggregation, Growth and Nucleation

If the breakage function and specific rate of breakage are set to zero :

$$\rho(v, w) = 0 \quad \forall\, v, w \tag{2.102}$$

$$S(v) = 0 \quad \forall\, v \tag{2.103}$$

then the PBE for combined aggregation and growth is obtained :

$$\frac{n(v) - n_{in}(v)}{\tau} + \frac{d}{dv}\Big(G(v)n(v)\Big) = \int_0^{v/2} \beta(v-w, w)n(v-w)n(w)dw - n(v)\int_0^{\infty} \beta(v, w)n(w)dw \tag{2.104}$$

In this simulation a particle free feed distribution was considered :

$$n_{in}(v) = 0 \qquad \forall\, v \in (0, \infty] \tag{2.105}$$

and additionally, the aggregation kernel and growth function were both taken to be constant :

$$\beta(v, w) = \beta_0 \tag{2.106}$$

$$G(v) = G_0 \tag{2.107}$$

The analytical solution to the resulting PBE (2.104) has been derived by Liao and Hulburt (1976) :

$$n(v) = 2n_0 \exp(-px)\frac{I_1(x)}{x} \tag{2.108}$$

where :

$$p = \sqrt{1 + \frac{1}{2\beta_0 n_0 G_0 \tau^2}} \tag{2.109}$$

$$x = \frac{v}{G_0}\sqrt{2\beta_0 n_0 G_0} \tag{2.110}$$

and :

$$n(0) = n_0 \tag{2.111}$$

is the number density of nuclei in the product stream.

The constants τ and n_0 were set to unity and simulations were performed with the values of β_0 and G_0 listed in Table (2.10).

If the PBE is multiplied by v^i and integrated over the domain $(0, \infty]$ then the following analytical expressions can be derived for the moments of the density distribution :

$$m_0 = \frac{-1 + \sqrt{1 + 2\beta_0 n_0 G_0 \tau^2}}{\beta_0 \tau} \tag{2.112}$$

$$m_1 = \tau G_0 m_0 \tag{2.113}$$

$$m_2 = \tau(\beta_0 m_1^2 + 2G_0 m_1) \tag{2.114}$$

The values of these moments are tabulated below for each of the problems investigated.

case	β_0	G_0	m_0	m_1	m_2
4a	1×10^{0}	1×10^{0}	7.321×10^{-1}	7.321×10^{-1}	$2.000 \times 10^{+0}$
4b	1×10^{1}	1×10^{1}	$1.318 \times 10^{+0}$	$1.318 \times 10^{+1}$	$2.000 \times 10^{+3}$
4c	1×10^{2}	1×10^{2}	$1.407 \times 10^{+0}$	$1.404 \times 10^{+2}$	$2.000 \times 10^{+6}$

Table 2.10: Analytical moments of the density distribution for the aggregation, growth and nucleation problem.

Truncation and refinement of the partition were performed as previously described for aggregation/growth dominant problems. The truncation points and the parameters of the refined geometric progression are shown in Table (2.11).

case	v^0_{max}	M_2^0	v_{max}	M_2	v_1	N
4a	2.500×10^{3}	0.9938	2.636×10^{1}	0.9991	0.5	20
4b	2.500×10^{5}	0.9858	2.379×10^{3}	0.9992	1.0	20
4c	2.500×10^{7}	0.9859	2.365×10^{5}	0.9992	1.0	40

Table 2.11: Domain truncation and partition parameters used in the combined aggregation, growth and nucleation problem.

The collocation-Galerkin formulation of the problem was solved using the successive substitution scheme (2.44). Log-log plots of the finite element solutions to the problem are shown in figure (2.10). Solid lines represent the analytical solution (2.108) and the symbols are the nodal values of the finite element solution for the first and last node of each element.

Errors in the moments of the finite element solution, mean square relative errors in the number of particles predicted in an interval, the number of equations solved, the CPU requirements of the simulations and the number of iterations required for convergence are all recorded below in Table (2.12).

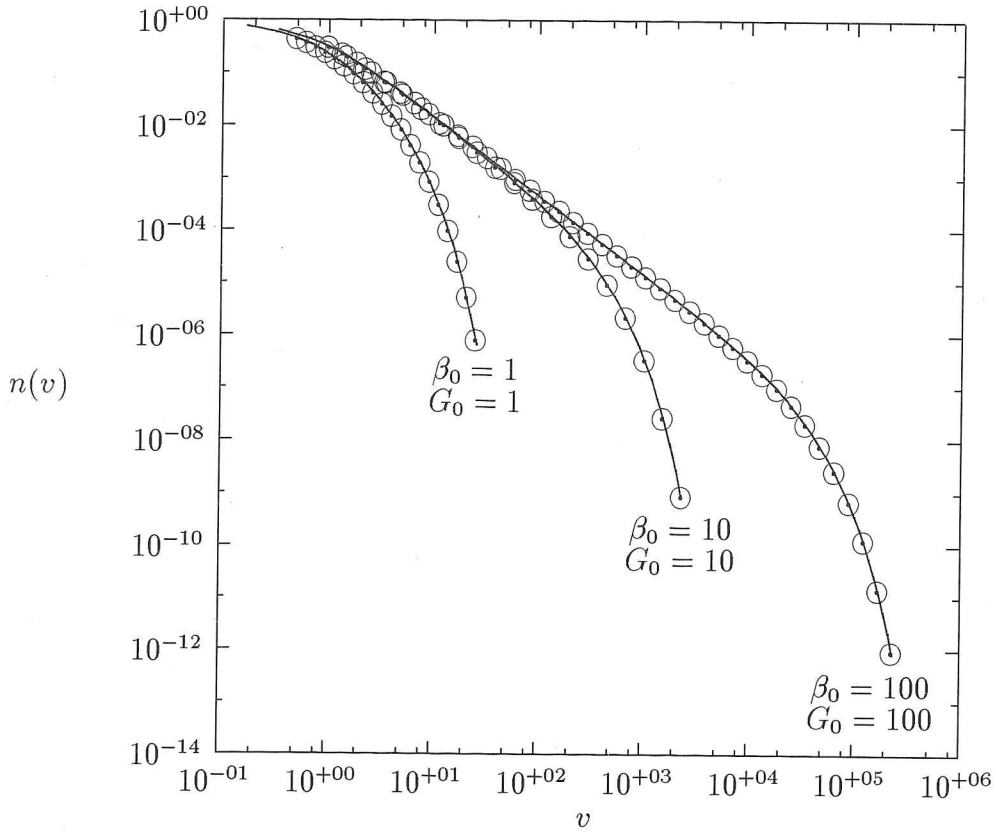

Figure 2.10: FEM predictions of the number density function for the aggregation, growth and nucleation problem with $n_{in}(v) = 0$, $\beta(v,w) = \beta_0$, $G(v) = G_0$ and $\tau = 1$ for $(\beta_0, G_0) = (1,1), (10,10)$ and $(100,100)$.

case	$E\ m_0$	$E\ m_1$	$E\ m_2$	$\frac{1}{N}SSRE$	N_{eq}	CPU	N_{it}
4a	0.055%	0.046%	0.013%	7.568×10^{-7}	61	$< 1u$	21
4b	0.217%	0.131%	0.141%	5.478×10^{-5}	61	$4u$	240
4c	0.203%	0.135%	0.200%	1.709×10^{-5}	121	$143u$	2253

Table 2.12: Errors in moments and mean square relative errors of the finite element solution to the aggregation, growth and nucleation problem.

In all cases excellent agreement between the numerically obtained density distribution and the analytical density distribution is quantified by small errors in the numerical predictions of moments of the density distribution and low $\frac{1}{N}$SSRE values.

2.15 A Comparison with the Performance of the DPB

In this section the performance of the finite element method is compared with that of the DPB. The DPB has been selected for this comparison since it was the only other method capable of solving the steady state PBE for aggregation, breakage, growth and nucleation with a general set of parameterizing functions.

The simulations for the aggregation problem of subsection 2.14.1 were repeated using the DPB of Litster, Smit and Hounslow (1995) as modified by Wynn (1996).

Prior to implementing this method the domain must be truncated to a non-zero lower limit ($0 < v_{min}$) as well as a finite upper limit ($v_{max} < \infty$). Hence, domain truncation errors are incurred at both the lower and upper limits of the domain. The lower limit was selected to be as large as possible such that the criterion :

$$\frac{\int_{v_{min}}^{\infty} n(v)dv}{\int_{0}^{\infty} n(v)dv} > 0.999 \tag{2.115}$$

was still satisfied.

In an attempt to match the accuracy of the finite element method, increasingly large discretization factors (q values) were used. Once the discretization factor was set, the number of equations used to approximate the system (N_{eq}) was selected to be as small as possible such that $M_2 > 0.999$. This in turn dictated the upper limit of the domain (v_{max}). When a discretization factor of $q = 24$ was reached, the resulting systems of equations became too large to solve with the available computer resources.

The parameters of the DPB simulations with a discretization factor of $q = 23$ are listed below in Table (2.13).

case	τ	v_{min}	v_{max}	N_{eq}
DPBa	2×10^2	0.001	3.526×10^3	502
DPBb	2×10^4	0.001	3.421×10^5	653
DPBc	2×10^6	0.002	3.646×10^7	781

Table 2.13: Domain truncation and partition parameters used for the DPB when solving the aggregation problem.

Figure (2.11) is a plot of the square relative error in the number of particles predicted in an interval by the DPB and the finite element method for case 1b of the aggregation problem. The square relative errors for the DPB appear as crosses while those of the finite element method appear as odots ($\odot$). Over most of the domain the SRE values of the finite element method are about two orders of magnitude smaller than those of the DPB of Litster, Smit and Hounslow (1995). A small region exists (in the vicinity of $v = 7 \times 10^4$) where near perfect error cancellations result in SRE values being smaller for the DPB.

Errors in the moments of the DPB solution, mean square relative errors, the number of equations comprising the system and the CPU times for each simulation are recorded below in Table (2.14). These values will be discussed in the following section.

case	$E\ m_0$	$E\ m_1$	$E\ m_2$	$\frac{1}{N}SSRE$	N_{eq}	CPU
DPBa	0.052%	0.000%	0.194%	2.591×10^{-4}	501	$216u$
DPBb	0.451%	0.000%	0.753%	4.034×10^{-5}	653	$469u$
DBPc	0.052%	0.000%	0.163%	7.101×10^{-5}	781	$731u$

Table 2.14: Errors in moments and mean square relative errors of the DPB solution to the aggregation problem.

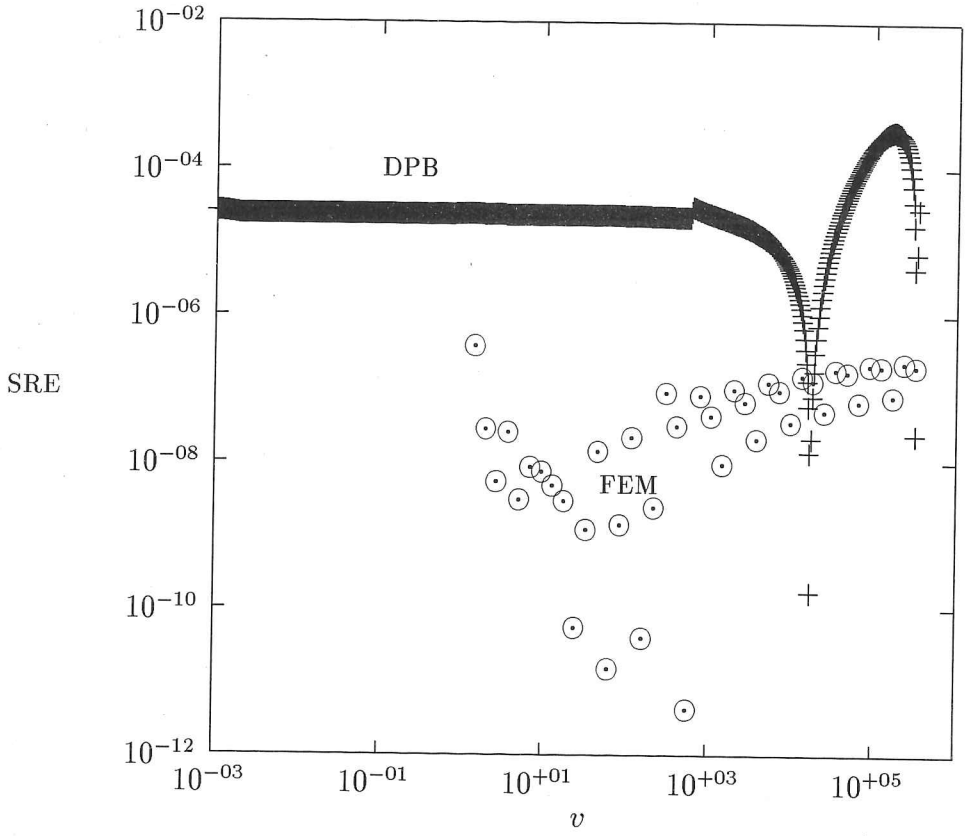

Figure 2.11: Square relative errors (SRE) for the finite element algorithm and the DPB of Litster, Smit and Hounslow (1995) in case 1b of the aggregation problem.

2.16 Discussion

The case studies sections 2.14.1–2.14.4 were selected to be very demanding tests on the finite element method's convergence capabilities and ability to predict the density distribution and moments correctly. Time constants, growth rates *etc.* were chosen so that density distribution of the output stream would be vastly different from that of the feed stream which was used to initialize the iterative procedures (2.43 and 2.44). The substantial changes in the feed distributions were quantified in the first two case studies by the indices of aggregation and breakage. In case 1c (see Table (2.1)) an index of aggregation of 0.999 was achieved. This is indicative of a thousand-fold reduction of the number of particles in the feed stream. Similar

changes in particle numbers can be observed for breakage in case 2c (see Table (2.4)), while in the growth example a million fold increase in the volume of particles of the feed stream was achieved (see Table (2.4)). Figures (2.6), (2.7), (2.9) and (2.10) show excellent agreement between the finite element solution and the analytical solutions for the problems investigated even in cases where significant changes in the density distribution occurred. This excellent agreement is quantified by the low $\frac{1}{N}$SSRE values listed in tables (2.3), (2.6), (2.9) and (2.12).

In order to predict the moments of the density distribution accurately, a method must be able to solve for a function with values that can change over many orders of magnitude. For instance the density distribution of case 1c assumes values between 0.286×10^{-18} and 0.500×10^{-3} over the domain of interest. In all cases the finite element method proved capable of this task. All moments were predicted to within one third of a percent of their analytical values.

Accurate prediction of moments requires appropriately chosen domain truncation points. Visual inspection of the $v^2 n_h(v)$ curve can be used to satisfy criterion (2.47) to within $\pm 0.1\%$. If additional accuracy is required the extrapolation procedure described in section 2.11 can be used to eliminate much of the time consuming guess-work associated with finding the upper limit by trial and error.

In simulation 1c, the finite element algorithm required over 16000 iterations to solve an aggregation problem. This suggests that the successive substitution algorithm is inefficient. However it was demonstrated by Nicmanis 1995 that in many cases less computational power was required by the successive substitution algorithm than by a Newton method applied to the same problem. Although the Newton method is capable of solving the problem in considerably fewer iterations, each iteration is much more expensive since a completely dense Jacobian matrix must be assembled. The successive substitution algorithm has the additional advantages of being much more robust and requiring significantly less computer memory. Both of these items will be discussed below.

In a direct comparison with the DPB it was shown that the finite element method was capable of more accurate predictions of the density distribution. In figure (2.11) it can be seen that the SRE values for the FEM are several orders of magnitude smaller than those of the DPB for case 1b. This fact is quantified by a comparison of the $\frac{1}{N}SSRE$ values in tables (2.3) and (2.14). The $\frac{1}{N}SSRE$ values are typically two to three orders of magnitude smaller for the finite element method.

The DPB algorithm is derived such that the zeroth and first moments of the density distribution are predicted correctly. In Table (2.14) it can be seen that the first moment is predicted exactly by the DPB. In all cases the finite element method predicted moments to within 0.2% of their analytical value and made slightly better predictions of the zeroth and second moments than the DPB. Comparable accuracy in predictions of these moments is achieved by the DPB (despite much larger SRE values) due to error cancellations.

Considerably less computational power is required by the finite element method. Tables (2.3 and 2.14) reveal that the finite element method converged using 1.85%, 13.6% and 78.4% of the computational power required by the DPB.

It should also be noted that the finite element method was initialized using the feed distribution which in many cases differed from the final converged solution by over ten orders of magnitude. The DPB algorithm would not converge from such poor initial guesses and was initialized from much better starting points that were within an order of magnitude of the final solution. In most practical applications such a good initial guess would not be available.

Figure (2.11) also highlights one of the shortcomings of the DPB. This method does not span the sub-domain $(0, v_1)$. Thus finite domain error is also incurred at the lower end of the volume domain. A small value of v_1 must be selected to reduce this finite domain error to an acceptably low level. Consequently the geometric progression

requires many intervals to span the lower volume range where the solution is changing very little across each element.

Computer memory requirements of the finite element method were significantly lower than those of the DPB for two reasons : i) the FEM is a higher order method and hence is capable of achieving better accuracy than the DPB using fewer equations (compare N_{eq} values of tables (2.3 and 2.14)) and ii) the successive substitution strategy retains the data in diagonal block form. A system of N_{eq} equations will result in a matrix of $\frac{N_{eq}-1}{p-1}$ overlapping $p \times p$ blocks, where $p-1$ is the order of the interpolation polynomial. The DPB uses Newton's method which requires an $N_{eq} \times N_{eq}$ Jacobian matrix. For instance in case 1c the finite element algorithm solve a system of equations with a matrix consisting of 45 4×4 blocks resulting in 719 non-zero matrix entries whereas the DPB solved a system of equations with a completely dense matrix consisting of $781 \times 781 = 609961$ non-zero entries.

2.17 Chapter Conclusions

The following conclusions can be made with regards to the finite element method developed in this chapter :

- The finite element method proved capable of accurately predicting the solutions to the steady state PBE and the moments of these solutions for the full range of particulate mechanisms (aggregation, breakage, growth and nucleation).
- The simple heuristic suggested in section 2.13 for refining the discretization according to a geometrical progression may be used to approach high accuracy solutions. However, this procedure requires a significant amount of guesswork and in many cases geometric discretizations are sub-optimal (as will be demonstrated in chapters 3 and 4).
- The quantity M_2 has been identified as one of central importance when calculating moments to very high accuracy (better than 0.1 %).
- A collocation-Galerkin formulation can be used to avoid the ill-conditioned matrices associated with growth problems.
- The method is derived without placing any restrictions upon the discretization of the domain hence this method is amenable to adaptive mesh refinement techniques that will be shown in chapter 4 to yield further increases in accuracy and computational efficiency.

The finite element method was found to have the following advantages over the DPB of Litster, Smit and Hounslow (1995) :

- Using smaller systems of equations to represent the PBE, the finite element method was found capable of making better predictions of the density distribution.

- The finite element method was found to be more robust and easier to converge and did not require any problem-specific scaling parameters.
- The finite element method does not incur any truncation errors at the lower volume range.
- No restrictions are placed on the location of grid points when using the finite element method. Hence, this method is amenable to adaptive mesh grading techniques whereas as the domain must be partitioned according to a geometrical progression when using the DPB.
- The finite element solution of the PBE can be a piecewise polynomial of any order whereas that of the DPB is assumed to be constant within each partition.
- The finite element method uses significantly less computer memory when solving the PBE.
- The finite element method requires significantly less CPU time to converge to a solution.

Chapter 3

An *a Posteriori* Error Estimate for the Steady State Population Balance Equation

In this chapter error estimates are derived to assess quantitatively the quality of obtained numerical solutions. These error estimates are successfully applied to several cases where the finite element method has been used to solve the steady state PBE. Discussion is presented on how these error estimates may be used to establish convergence of the finite element solution to the analytical solution of certain classes of problems. This chapter concludes with consideration of the potential use of these error estimates in a rigorous approach to automatic mesh refinement.

3.1 What is an Error Estimate ?

Error estimates are inequalities that place an upper bound on the "distance" between the analytical solution and a numerically obtained solution to a problem. *A priori* estimates for finite element methods typically use properties of the variational formulation of the problem while *a posteriori* estimates also make use of the obtained numerical solution. This chapter will focus upon *a posteriori* estimates since they are acknowledged by Zeidler (1995) to yield tighter error bounds.

3.2 Objectives of Deriving an Error Estimate

One motivation for deriving error bounds for the numerical solutions to the steady state PBE has already been stated above; namely that error estimates are useful in quantifying the quality of an obtained numerical solution. Error estimates, however, typically yield much more information than "how close is a numerical solution to the analytical solution". They may also be used to locate regions where very large or very small errors in the numerical solution exist. Hence rather than refining every element composing a domain (as in the refinement procedure of chapter 2) with the assistance of an error estimate we may instead deduce which individual elements should be refined. The error estimate derived in this chapter will lie at the core of an automatic mesh refinement algorithm to be described in chapter 4.

3.3 Literature Review

At the time of writing, no error estimate had yet been proposed for the steady state population balance equation. In fact error estimates for nonlinear problems are currently quite scarce. Existing developments in this area tend to concentrate upon

general classes of problems and are largely devoted to establishing the convergence of a particular finite element method to an analytical solution. A notable example is the work of Zenisek (1990) who ties together the earlier developments of Ciarlet (1978), Ciarlet, Schultz and Varga (1969), Melkes (1970) and Noor and Whiteman (1976) to derive an error estimate for the finite element formulation of a two dimensional, second order, nonlinear, partial differential equation. A set of assumptions was introduced to ensure the variational form of the problem was strongly monotonic and Lipschitz continuous. These two properties were then used to derive the estimate. Although this estimate adequately establishes convergence of the method it is derived in terms of parameters that are difficult to calculate, hence modifications are necessary if this estimate is to yield quantitative predictions and be of potential use in an automatic refinement algorithm.

The derivation of section 3.6 is based upon Zenisek's proof, however a different set of assumptions is introduced so that the final error estimate is derived in terms of calculable parameters. Additional modifications are necessary since the PBE is a first order equation (not second order) and the finite element method derived in chapter 2 uses piecewise Lagrange and not piecewise Hermite polynomials.

An error estimate will first be derived for a general class of problems and then in section 3.8 this estimate will be applied to the steady state PBE.

3.4 Notation

More concise notation will be introduced in this section and the finite element method will be reformulated in terms of this notation for a general class of problems. Attention should also be brought to the fact that in sections 3.4-3.7 the symbols "u" and "v" represent functions while "x" represents the domain of the independent variable. More specifically the function "u" refers to the analytical solution of the stated prob-

lem while "v" is typically an arbitrary element of a function space to which "u" also belongs. This notation has been adopted in compliance with the standard notation of functional analysis. In section 3.8 however the notation of Litster, Smit and Hounslow (1995) for particulate systems will be adopted. According to this notation, the symbol "n" will represent the analytical solution to a problem and "v" will be the domain of the independent variable (and not a function as defined in earlier sections).

Let Ω be the interval (a, b) and $C(\Omega)$ be the space of continuous functions over Ω. Now consider the problem of finding a function $u(x) \epsilon C(\Omega)$ such that :

$$q[x, u(x), u'(x)] = f(x) \quad \forall\, x \epsilon \Omega \tag{3.1}$$

where $q[x, u(x), u'(x)]$ is a nonlinear operator on the function $u(x)$ and its first derivative $u'(x)$. If the above expression is multiplied by a weight function $\phi(x) \epsilon C(\Omega)$ and integrated over the domain $\Omega = (a, b)$ then (3.1) may be reformulated as the weighted residual problem of finding a function $u(x) \epsilon C(\Omega)$ such that :

$$a(u, \phi) = L(\phi) \quad \forall\, \phi(x) \epsilon C(\Omega) \tag{3.2}$$

where :

$$a(u, \phi) = \int_\Omega \phi(x) q[x, u(x), u'(x)] d\Omega \tag{3.3}$$

$$L(\phi) = \int_\Omega \phi(x) f(x) d\Omega \tag{3.4}$$

In the finite element formulation of this problem the solution is approximated by $u_h(x) \epsilon V_h(\Omega)$, where $V_h(\Omega) \subset C(\Omega)$ is defined :

$$V_h(\Omega) = \left\{ \begin{array}{ll} v_h(x) : & v_h(x) \text{ is a piecewise continuous polynomial with possible} \\ & \text{discontinuities in its derivatives at the inter-element} \\ & \text{boundaries.} \end{array} \right\}$$

Then the problem becomes one of finding a function $u_h(x) \epsilon V_h(\Omega)$ such that :

$$a(u_h, \phi) = L(\phi) \quad \forall\, \phi(x) \epsilon V_h(\Omega) \tag{3.5}$$

3.5 Assumptions

The following three assumptions will be used in the derivation of the error estimate.

(A1) Assume that there exists a constant $\alpha > 0$ such that :

$$\alpha \left(\left\| \eta_1 \right\|_\Omega^2 + \left\| \eta_2 \right\|_\Omega^2 \right) \leq \left| \int_\Omega \eta_1^2 \gamma_1 + \eta_1 \eta_2 \gamma_2 dx \right| \quad \forall\, \eta_1, \eta_2, \gamma_1, \gamma_2 \epsilon C(\Omega)$$

where $\|\bullet\|_\Omega$ denotes the standard L_2-norm over the domain Ω. The L_2-norm (as defined for instance by Adams (1975)) will be used throughout this chapter as a measure of the "distance" between two functions.

(A2) Assume that there exists a constant $K_1 < \infty$ such that :

$$\left| q_2[x, w, w'] \right| \leq K_1 \quad \forall\, w \epsilon C(\Omega)$$

(A3) Assume that there exists a constant $K_2 < \infty$ such that :

$$\left| q_3[x, w, w'] \right| \leq K_2 \quad \forall\, w \epsilon C(\Omega)$$

where the subscripts $_2$ and $_3$ denote partial differentiation of the operator q with respect to the second and third arguments.

3.6 Derivation of the Error Estimate

The consequences of four lemmas will be used in the final derivation of the error estimate. In lemma 1, assumption **(A1)** will be used to establish a property analogous

to strong monotonicity of the form $a(u,\phi)$ while in lemma 2 assumptions (**A2**) and (**A3**) will be used to infer Lipschitz continuity of this form. These two lemmas will be combined in lemma 3 to bound the norm $\|u_h - v\|$ where v is an arbitrary piecewise continuous polynomial. The function v will eventually be set to a perfect interpolant of the analytical solution, the properties of which are derived in lemma 4. The final theorem results from an application of the triangle inequality and lemma 3.

Lemma 1. *For all $u \in C(\Omega)$ and $v \in V_h(\Omega)$ there exists a constant $\alpha > 0$ such that :*

$$\boxed{\alpha\left(\left\|u-v\right\|_{\Omega}^{2}+\left\|u'-v'\right\|_{\Omega}^{2}\right) \leq \left|a(u,u-v)-a(v,u-v)\right|}$$

Proof: Define $g(t) := q[x, v + t(u-v), v' + t(u'-v')]$ then :

$$\begin{aligned} g(1) - g(0) &= \int_0^1 g'(T)dT \\ &= \int_0^1 (u-v)q_2[x, v+T(u-v), v'+T(u'-v')] \\ &+ (u'-v')q_3[x, v+T(u-v), v'+T(u'-v')]dT \\ &= q[x,u,u'] - q[x,v,v'] \end{aligned} \tag{3.6}$$

Multiplying by $(u-v)$ and integrating over the domain Ω :

$$\left|\int_\Omega \int_0^1 (u-v)^2 q_2 + (u-v)(u'-v')q_3 \, dT dx\right| = \left|a(u,u-v) - a(v,u-v)\right| \tag{3.7}$$

where the arguments of the operator q have been neglected in order to simplify notation. By enforcing assumption (**A1**) the desired result is obtained :

$$\alpha\left(\left\|u-v\right\|_{\Omega}^{2}+\left\|u'-v'\right\|_{\Omega}^{2}\right) \leq \left|a(u,u-v)-a(v,u-v)\right| \qquad \square$$

Lemma 2. *Let $u, w \in C(\Omega)$ and $v \in V_h$ then there exist constants $K_1, K_2 < \infty$ such that the following inequality may be satisfied :*

$$\boxed{\left|a(u,w) - a(v,w)\right| \leq \left\|w\right\|_{\Omega} \left(K_1 \left\|u - v\right\|_{\Omega} + K_2 \left\|u' - v'\right\|_{\Omega}\right)}$$

Proof: As in the previous proof the following function is defined :

$$g(t) := q[x, v + t(u - v), v' + t(u' - v')] \tag{3.8}$$

Expression (3.6) is then considered :

$$\begin{aligned} q[x,u,u'] - q[x,v,v'] = \int_0^1 (u-v) q_2[x, v + T(u-v), v' + T(u'-v')] \\ + (u'-v') q_3[x, v + T(u-v), v' + T(u'-v')] dT \end{aligned} \tag{3.9}$$

The above expression is then multiplied by a function w : $w \in C(\Omega)$ and integrated over the domain Ω to obtain :

$$\begin{aligned} \left|a(u,w) - a(v,w)\right| &= \left|\int_\Omega \left(w(u-v) \int_0^1 q_2 \, dT + w(u'-v') \int_0^1 q_3 \, dT \right) dx \right| \\ &\leq \int_\Omega \left(\left|w(u-v)\right| \int_0^1 \left|q_2\right| \, dT + |w(u'-v')| \int_0^1 \left|q_3\right| \, dT \right) dx \end{aligned} \tag{3.10}$$

Assumptions **(A2)** and **(A3)** are applied to obtain :

$$\left|a(u,w) - a(v,w)\right| \leq K_1 \int_\Omega \left|w(u-v)\right| dx + K_2 \int_\Omega w(u'-v') \, dx \tag{3.11}$$

Application of the Cauchy-Schwartz inequality to the above expression yields the desired result :

$$\left|a(u,w) - a(v,w)\right| \leq \left\|w\right\|_{\Omega} \left(K_1 \left\|u - v\right\|_{\Omega} + K_2 \left\|u' - v'\right\|_{\Omega}\right) \qquad \square$$

Lemma 3. *Let $u \in C(\Omega)$ be the solution of the weighted residual problem (3.2), let $u_h \in V_h(\Omega)$ be the solution of the finite element problem (3.5) and let v be an arbitrary element of V_h then there exist constants $0 < K_1, K_2, \alpha, c_1 < \infty$ such that :*

$$\boxed{\left\|u_h - v\right\|_\Omega \leq \frac{1}{\alpha(1+c_1^2)}\left(K_1\left\|u - v\right\|_\Omega + K_2\left\|u' - v'\right\|_\Omega\right) \qquad \forall\, v \in V_h(\Omega)}$$

Proof : Beginning with lemma 1 $\forall\, u \in C(\Omega)$ and $\forall\, v \in V_h(\Omega)$ there exists $\alpha > 0$ such that :

$$\alpha\left(\left\|u - v\right\|_\Omega^2 + \left\|u' - v'\right\|_\Omega^2\right) \leq \left|a(u, u-v) - a(v, u-v)\right| \tag{3.12}$$

Since $u_h \in V_h(\Omega)$ then $u_h \in C(\Omega)$. Thus setting $u = u_h$ in the above expression :

$$\alpha\left(\left\|u_h - v\right\|_\Omega^2 + \left\|u_h' - v'\right\|_\Omega^2\right) \leq \left|a(u_h, u_h - v) - a(v, u_h - v)\right| \quad \forall\, v \in V_h(\Omega) \tag{3.13}$$

Since u_h is the solution to the problem (3.5) then $a(u_h, u_h - v) = L(u_h - v)$, hence :

$$\alpha\left(\left\|u_h - v\right\|_\Omega^2 + \left\|u_h' - v'\right\|_\Omega^2\right) \leq \left|L(u_h - v) - a(v, u_h - v)\right| \quad \forall\, v \in V_h(\Omega) \tag{3.14}$$

Subtracting and then adding $a(u, u_h - v)$:

$$\alpha\left(\left\|u_h - v\right\|_\Omega^2 + \left\|u_h' - v'\right\|_\Omega^2\right) \leq \left|L(u_h - v) - a(u, u_h - v) + a(u, u_h - v) - a(v, u_h - v)\right| \tag{3.15}$$

And if u is the solution to the problem (3.2) then $a(u, u_h - v) = L(u_h - v)$ hence :

$$\alpha\left(\left\|u_h - v\right\|_\Omega^2 + \left\|u_h' - v'\right\|_\Omega^2\right) \leq \left|a(u, u_h - v) - a(v, u_h - v)\right| \quad \forall\, v \in V_h(\Omega) \tag{3.16}$$

Since $u, u_h - v \in C(\Omega)$ and $v \in V_h \subset C(\Omega)$ then lemma 2 may be applied to the right hand side of the above expression :

$$\alpha\left(\left\|u_h - v\right\|_\Omega^2 + \left\|u_h' - v'\right\|_\Omega^2\right) \leq \left\|u_h - v\right\|_\Omega \left(K_1 \left\|u - v\right\|_\Omega + K_2 \left\|u' - v'\right\|_\Omega\right)$$

$$\forall\, v \in V_h(\Omega) \quad (3.17)$$

For each $v \in V_h(\Omega)$ let a constant c_1 be defined :

$$c_1 = \frac{\left\|u_h' - v'\right\|_\Omega}{\left\|u_h - v\right\|_\Omega} \tag{3.18}$$

The desired inequality is obtained by substituting $\|u_h' - v'\|_\Omega = c_1\|u_h - v\|_\Omega$ into expression (3.17) and dividing through by $\alpha(1 + c_1^2)\|u_h - v\|_\Omega$:

$$\left\|u_h - v\right\|_\Omega \leq \frac{1}{\alpha(1 + c_1^2)}\left(K_1 \left\|u - v\right\|_\Omega + K_2 \left\|u' - v'\right\|_\Omega\right) \quad \forall\, v \in V_h(\Omega) \qquad \square$$

Lemma 4. *Let u be the solution to the weighted residual problem (3.2) within element e of length h_e. It is also assumed that within each element u_h^e is a Lagrange polynomial of order p. Now let πu be the perfect linear interpolant of the solution u of the problem (3.2), ie. $\pi u(x_i) = u(x_i)$ at each of the $p+1$ nodes ($x_i's$), and is linear in-between the nodes, then i) :*

$$\boxed{\left\|u - \pi u\right\|_{\Omega_e} \leq \frac{h_e^{\frac{5}{2}}}{p^2\sqrt{120}} \max\left|u''(\xi)\right|_{\xi \in \Omega_e}}$$

and furthermore ii) :

$$\boxed{\left\|u' - \pi' u\right\|_{\Omega_e} \leq \frac{h_e^{\frac{3}{2}}}{p\sqrt{12}} \max\left|u''(\xi)\right|_{\xi \in \Omega_e}}$$

where $\pi' u$ denotes the slope of the linear interpolant πu.

Proof of i) : It is known from interpolation theory (see for instance Gerald and Wheatly (1989)) that :

$$\left|u - \pi u\right| \leq \left|\frac{(x - x_1)(x - x_2)}{2}\right| \max \left|u''(\xi)\right|_{\xi \in \Omega_e} \tag{3.19}$$

where x_1 and x_2 are two adjacent nodal co-ordinates between which the unknown function is approximated by a line. Substituting this result into the definition of the norm :

$$\left\|u - \pi u\right\|_{\Omega_e} \leq \left(p \int_{x_1}^{x_2} \left(\frac{(x - x_1)(x - x_2)}{2} \max \left|u''(\xi)\right|_{\xi \in \Omega_e} \right)^2 dx \right)^{\frac{1}{2}} \tag{3.20}$$

In the above case p adjacent linear sections are used to approximate the function u. Integration yields the following simplification :

$$\left\|u - \pi u\right\|_{\Omega_e} \leq \left(p \frac{(x_2 - x_1)^5}{120} \right)^{\frac{1}{2}} \max \left|u''(\xi)\right|_{\xi \in \Omega_e} \tag{3.21}$$

but $x_2 - x_1 = \frac{h_e}{p}$, hence :

$$\left\|u - \pi u\right\|_{\Omega_e} \leq \frac{h_e^{\frac{5}{2}}}{p^2 \sqrt{120}} \max \left|u''(\xi)\right|_{\xi \in \Omega_e} \qquad \square$$

Proof of ii) follows similarly after differentiating expression (3.19) with respect to x.

The Error Estimate. *Let $u(x)$ be the solution of the weighted residual problem (3.2) and let $u_h(x)$ be the solution of the finite element problem (3.5). Then :*

$$\boxed{\left\|u - u_h\right\|_{\Omega_e} \leq \left[\left(1 + \frac{K_1}{\alpha(1 + c_1^2)} \right) \frac{h_e^{\frac{5}{2}}}{p^2 \sqrt{120}} + \frac{K_2}{\alpha(1 + c_1^2)} \frac{h_e^{\frac{3}{2}}}{p\sqrt{12}} \right] \max \left|u''(\xi)\right|_{\xi \in \Omega_e}} \quad (\dagger)$$

Proof :

$$\begin{aligned} \left\|u - u_h\right\|_{\Omega} &= \left\|(u - v) + (v - u_h)\right\|_{\Omega} \\ &\leq \left\|u - v\right\|_{\Omega} + \left\|u_h - v\right\|_{\Omega} \end{aligned} \tag{3.22}$$

where $v(x)$ is an arbitrary element of $V_h(\Omega)$.

Applying lemma 3 to the second term on the right hand side of the above inequality :

$$\begin{aligned}\left\|u-u_h\right\|_\Omega &\leq \left\|u-v\right\|_\Omega + \frac{1}{\alpha(1+c_1^2)}\left(K_1\left\|u-v\right\|_\Omega + K_2\left\|u'-v'\right\|_\Omega\right) \\ &= \left(1+\frac{K_1}{\alpha(1+c_1^2)}\right)\left\|u-v\right\|_\Omega + \frac{K_2}{\alpha(1+c_1^2)}\left\|u'-v'\right\|_\Omega \quad \forall\, v \,\epsilon\, V_h(\Omega)\end{aligned} \tag{3.23}$$

If we consider the above inequality over the e^{th} sub-domain Ω_e and select v to be the perfect linear interpolant πu then lemmas 4i and 4ii may be applied to the above expression to obtain the desired error estimate in terms of the element length :

$$\left\|u-u_h\right\|_{\Omega_e} \leq \left[\left(1+\frac{K_1}{\alpha(1+c_1^2)}\right)\frac{h_e^{\frac{5}{2}}}{p^2\sqrt{120}} + \frac{K_2}{\alpha(1+c_1^2)}\frac{h_e^{\frac{3}{2}}}{p\sqrt{12}}\right]\max\left|u''(\xi)\right|_{\xi\,\epsilon\,\Omega_e} \qquad \square$$

3.7 A Simplified Error Estimate for No-Flux Problems

A simplified version of the error estimate may be derived when the operator q is a function of the independent variable x and the unknown function u only. Consider the problem of finding a function $u(x)\,\epsilon\,C(\Omega)$ such that :

$$q[x,u(x)] = f(x) \quad \forall\, v \,\epsilon\, \Omega \tag{3.24}$$

In this case we make the assumptions :

(A4) Assume that there exists a constant $\lambda > 0$ such that :

$$\lambda\left\|\eta\right\|_\Omega^2 \leq \left|\int_\Omega \eta^2\gamma dx\right| \quad \forall\, \eta,\gamma \,\epsilon\, C(\Omega) \tag{3.25}$$

(A5) Assume that there exists a constant $K_3 < \infty$ such that :

$$\left| q_2[x,w] \right| \leq K_3 \quad \forall\, w \in C(\Omega) \tag{3.26}$$

Using these two assumptions the following error estimate may be derived for no-flux problems :

$$\boxed{\left\| u - u_h \right\|_{\Omega_e} \leq \left(1 + \frac{K_3}{\lambda}\right) \frac{h_e^{\frac{5}{2}}}{p^2\sqrt{120}} \max \left| u''(\xi) \right|_{\xi \in \Omega_e}} \qquad (\ddagger)$$

3.8 Application of the Estimate to the Population Balance Equation

If we set :

$$q[v, n(v), n'(v)] = n(v) \left[1 + \tau \left(\frac{dG(v)}{dv} + \bar{d}[v, n(v)] \right) \right]$$
$$- \tau \left(b[v, n(v)] - G(v) \frac{dn(v)}{dv} \right) \tag{3.27}$$

and :

$$f(v) = n_{in}(v) \tag{3.28}$$

then the PBE is of the form (3.1) and the error estimate (†) may be applied to this problem. The notation for $\bar{d}[v, n(v)]$ and $b[v, n(v)]$ has been previously declared in expressions 2.8 and 2.11.

Alternatively, if we set :

$$q[v, n(v)] = n(v) \left[1 + \tau \bar{d}[v, n(v)] \right] - \tau b[v, n(v)] \tag{3.29}$$

then the PBE is of the form (3.24) and the simplified error estimate (‡) may be applied.

3.9 Estimation of the Constants

Means of approximating the constants $\alpha, \lambda, c_1, K_1, K_2,$ and, K_3 will be presented in this section. Once these values are obtained the error estimate may be used to assess quantitatively the quality of an obtained numerical solution. Approximations will be made in such a way that the errors in the error estimate will tend to zero as the numerical solution tends towards the analytical solution.

3.9.1 Estimation of α_e and λ_e

The derivation of the error estimate requires the following inequality to be valid within each element :

$$\alpha_e \left(\left\| n_h - \pi n \right\|_{\Omega_e}^2 + \left\| n_h' - \pi' n \right\|_{\Omega_e}^2 \right) \leq \left| \int_{\Omega_e} (n_h - \pi n)^2 \int_0^1 q_2 \, dT dv + \int_{\Omega_e} (n_h - \pi n)(n_h' - \pi' n) \int_0^1 q_3 \, dT dv \right| \quad (3.30)$$

where : $q_i = q_i[v, \pi n + T(n_h - \pi n), \pi' n + T(n_h' - \pi' n)]$ for $i = 2$ and 3, n_h is the finite element solution to the problem, πn is a perfect polynomial interpolant of the analytical solution to the problem and $\pi' n$ is the gradient of this interpolant.

The integral of q_2 with respect to T will be approximated by a trapezoidal rule weighting :

$$\int_0^1 q_2 \, dT \approx \frac{q_2[v, n_h, n_h'] + q_2[v, \pi n, \pi' n]}{2} \quad (3.31)$$

and similarly :

$$\int_0^1 q_3 \, dT \approx \frac{q_3[v, n_h, n_h'] + q_3[v, \pi n, \pi' n]}{2} \quad (3.32)$$

The above two approximations are substituted into (3.30) which is then divided by the term $\|n_h - \pi n\|^2_{\Omega_e} + \|n'_h - \pi' n\|^2_{\Omega_e}$ to obtain :

$$\alpha_e \leq \frac{1}{\|n_h - \pi n\|^2_{\Omega_e} + \|n'_h - \pi' n\|^2_{\Omega_e}} \times \left| \int_{\Omega_e} (n_h - \pi n)^2 \frac{q_2[v, n_h, n'_h] + q_2[v, \pi n, \pi' n]}{2} + (n_h - \pi n)(n'_h - \pi' n) \frac{q_3[v, n_h, n'_h] + q_3[v, \pi n, \pi' n]}{2} dv \right| \quad (3.33)$$

The functions πn and $\pi' n$ cannot be determined without knowing the analytical solution to the problem. Hence, these functions are approximated by πn_h and $\pi' n_h$ which tend towards πn and $\pi' n$ as n_h tends towards n. After making these approximations we set :

$$\alpha_e := \frac{1}{\|n_h - \pi n_h\|^2_{\Omega_e} + \|n'_h - \pi' n_h\|^2_{\Omega_e}} \left| \int_{\Omega_e} (n_h - \pi n_h)^2 \frac{q_2[v, n_h, n'_h] + q_2[v, \pi n_h, \pi' n_h]}{2} + (n_h - \pi n_h)(n'_h - \pi' n_h) \frac{q_3[v, n_h, n'_h] + q_3[v, \pi n_h, \pi' n_h]}{2} dv \right| \quad (3.34)$$

In no-flux problems $q = q[x, u]$ hence $q_3 = 0$ we therefore analogously define the parameter λ_e :

$$\lambda_e := \frac{1}{\|n_h - \pi n_h\|^2_{\Omega_e}} \left| \int_{\Omega_e} (n_h - \pi n_h)^2 \frac{q_2[v, n_h] + q_2[v, \pi n_h]}{2} dv \right| \quad (3.35)$$

3.9.2 Estimation of K_1^e, K_2^e and K_3^e

The derivation of the error estimate requires that within each element :

$$\left| q_2[v, \pi n + T(n_h - \pi n), \pi' n + T(n'_h - \pi' n)] \right| \leq K_1^e \quad \forall\, T \in (0,1), \quad \forall\, v \in \Omega_e \quad (3.36)$$

Once again πn and $\pi' n$ are approximated by the functions πn_h and $\pi' n_h$. Since πn_h is a perfect interpolant of n_h it is reasonable to assume that if we set :

$$K_1^e = \max \left| q_2[v, n_h, n'_h] \right|_{v \in \Omega_e} \quad (3.37)$$

then the inequality (3.36) will be satisfied. The operator q_2 may be evaluated using the expressions (3.45)–(3.53).

Over most elements, q_2 may be observed to be either monotonically increasing or decreasing with respect to v, hence we set :

$$K_1^e := \max\left(\left|q_2[v_a^e, n_h, n_h']\right|, \left|q_2[v_b^e, n_h, n_h']\right|\right) \tag{3.38}$$

and similarly we also set :

$$K_2^e := \max\left(\left|q_3[v_a^e, n_h, n_h']\right|, \left|q_3[v_b^e, n_h, n_h']\right|\right) \tag{3.39}$$

where the operator q_3 is evaluated using expression (3.53).

In no-flux problems we set :

$$K_3^e := \max\left(\left|q_2[v_a^e, n_h]\right|, \left|q_2[v_b^e, n_h]\right|\right) \tag{3.40}$$

3.9.3 Estimation of c_1^e

The following inequality was used in the derivation of the error estimate :

$$c_1 \left\| n_h - \pi n \right\|_{\Omega_e} \leq \left\| n_h' - \pi' n \right\|_{\Omega_e} \tag{3.41}$$

Again πn and $\pi' n$ are approximated by πn_h and $\pi' n_h$. We then set :

$$c_1^e := \frac{\left\| n_h' - \pi' n_h \right\|_{\Omega_e}}{\left\| n_h - \pi n_h \right\|_{\Omega_e}} \tag{3.42}$$

3.9.4 An Expression for $q_2[v, n(v), n'(v)]$

In order to estimate K_1, K_2 and K_3 we must have some means of evaluating the partial derivatives of the operator q. Whereas q_1 and q_3 may be determined by direct

partial differentiation of the PBE, the determination of q_2 is less straight-forward due to the presence of the integral expressions. An expression for q_2 may be indirectly obtained by ordinary differentiation of equation (3.1) with respect to v :

$$\frac{d}{dv}\left(q[v, n(v), n'(v)]\right) = \frac{df(v)}{dv} \tag{3.43}$$

Applying the chain rule for differentiation to the left-hand side of the above equation we obtain :

$$q_1[v, n(v), n'(v)] + q_2[v, n(v), n'(v)]\frac{dn(v)}{dv} + q_3[v, n(v), n'(v)]\frac{d^2n(v)}{dv^2} = \frac{df(v)}{dv} \tag{3.44}$$

An expression for $q_2[v, n(v), n'(v)]$ may be obtained by rearrangement of (3.44) :

$$q_2[v, n(v), n'(v)] = \frac{\dfrac{df(v)}{dv} - \left(q_1[v, n(v), n'(v)] + q_3[v, n(v), n'(v)]\right)}{\dfrac{dn(v)}{dv}} \tag{3.45}$$

In no-flux problems $q = q[v, n(v)]$ hence the above expression may be simplified as follows :

$$q_2[v, n(v)] = \frac{\dfrac{df(v)}{dv} - q_1[v, n(v)]}{\dfrac{dn(v)}{dv}} \tag{3.46}$$

3.9.5 An Expression for $q_1[v, n(v), n'(v)]$

Partial differentiation of the operator $q[v, n(v), n'(v)]$ with respect to v yields an expression for $q_1[v, n(v), n'(v)]$:

$$\begin{aligned} q_1[v, n, n'] &= \frac{\partial}{\partial v}\left(n\left[1 + \tau\left(\frac{dG(v)}{dv} + \bar{d}[v, n]\right)\right] - \tau\left(b[v, n] + G(v)\frac{dn}{dv}\right)\right) \\ &= \tau n\left(\frac{d^2G(v)}{dv^2} + \bar{d}_1[v, n]\right) - \tau\left(b_1[v, n] + \frac{dG(v)}{dv}\frac{dn}{dv}\right) \end{aligned} \tag{3.47}$$

The terms $\bar{d}[v,n]$ and $b[v,n]$ have been previously defined in equations (2.8) and (2.11).

In no-flux problems $q = q[v,n(v)]$ and $G(v) = 0$ hence the above expression may be simplified as follows :

$$q_1[v,n] = \tau\left(n\bar{d}_1[v,n] - b_1[v,n]\right) \tag{3.48}$$

In equations (3.47) and (3.48) the arguments of n and n' have been suppressed to emphasize that partial differentiation is being performed with respect to v while n and n' are held constant.

The integral terms are partially differentiated as follows :

$$\begin{aligned}\bar{d}_1[v,n] &= \frac{\partial}{\partial v}\left(\int_0^\infty \beta(v,w)n(w)dw + S(v)\right) \\ &= \int_0^\infty \beta_1(v,w)n(w)dw + \frac{dS(v)}{dv}\end{aligned} \tag{3.49}$$

$$b_1[v,n] = \frac{\partial}{\partial v}\left(b^a(v) + b^b(v)\right) = \frac{\partial}{\partial v}\left(b^a(v)\right) + \frac{\partial}{\partial v}\left(b^b(v)\right) \tag{3.50}$$

where $b^a(v)$ and $b^b(v)$ have been defined in equations (2.12) and (2.13) and may be partially differentiated as follows :

$$\begin{aligned}\frac{\partial}{\partial v}\left(b^a(v)\right) &= \frac{\partial}{\partial v}\int_0^{\frac{v}{2}} \beta(v-w,w)n(v-w)n(w)dw \\ &= \frac{1}{2}\beta\left(\frac{v}{2},\frac{v}{2}\right)\left[n\left(\frac{v}{2}\right)\right]^2 + \int_0^{\frac{v}{2}} \beta_1(v-w,w)n(v-w)n(w)dw\end{aligned} \tag{3.51}$$

and :

$$\begin{aligned}\frac{\partial}{\partial v}\left(b^b(v)\right) &= \frac{\partial}{\partial v}\int_v^\infty \rho(v,w)S(w)n(w)dw \\ &= -\rho(v,v)S(v)n(v) + \int_v^\infty \rho_1(v,w)S(w)n(w)dw\end{aligned} \tag{3.52}$$

3.9.6 An Expression for $q_3[v, n(v), n'(v)]$

Partial differentiation of $q[v, n, n']$ (as defined in (3.27)) with respect to n' yields :

$$q_3[v, n, n'] = \tau G(v) \tag{3.53}$$

3.9.7 Estimation of $\max |n''(\xi)|_{\xi \in \Omega_e}$

Once again $n(v)$ is approximated by $n_h(v)$ and we then assume that $n_h''(v)$ is monotonically increasing or decreasing throughout most elements and set :

$$\max \left| n''(\xi) \right|_{\xi \in \Omega_e} := \max \left(\left| n_h''(v_a^e) \right|, \left| n_h''(v_b^e) \right| \right) \tag{3.54}$$

3.10 Numerical Case Studies

Estimates of the error were made for each of the numerical simulations investigated in section 2.14. These estimates were compared with the actual errors present in the numerical solutions. It should be noted that the error estimates (†) and (‡) are derived for the Galerkin formulation of the problem while the numerical results of chapter 2 were obtained using a collocation or a collocation-Galerkin formulation. In each case the simulations were repeated. Initial convergence was obtained using a collocation or collocation-Galerkin formulation and then this solution was re-converged using a Galerkin formulation. Domain truncation, discretization and refinement were all performed as described previously.

3.10.1 Quantifying the Error

The following relative error norm (REN) is defined for each element e as the L_2-norm divided by $\bar{n}_h^e h_e^{1/2}$:

$$REN_e := \frac{\|n - n_h\|_{\Omega_e}}{\bar{n}_h^e h_e^{1/2}} = \frac{1}{\bar{n}_h^e h_e^{1/2}} \left(\int_{v_a^e}^{v_b^e} (n(v) - n_h(v))^2 dv \right)^{1/2} \tag{3.55}$$

where : $\bar{n}_h^e$ is the value of the finite element solution at the mid-point of element e. This norm is a quantitative measure of the discrepancy between the analytical solution and the numerically obtained solution to a problem. It will be used to test the ability of estimates (†) and (‡) to predict upper bounds of the error present in obtained numerical solutions. Division of the L_2-norm by a mean value of the solution within the element $\bar{n}_h^e$ has been performed to convert it to a relative quantity. As such it will be more useful when assessing errors in functions with values that typically change by several orders of magnitude over the domain of interest. The $h_e^{1/2}$ term is present in the denominator to average the error over the element length. Hence, elements containing segments of analytical and numerical solutions that are on average the same "distance" apart (in the relative, square integral sense) will have REN values of similar magnitudes regardless of their lengths.

We also define a relative error estimate (REE) for each element by dividing the error estimate (†) by $\bar{n}_h^e h_e^{1/2}$:

$$REE_e := \frac{1}{\bar{n}_h^e h_e^{1/2}} \left[\left(1 + \frac{K_1}{\alpha(1+c_1^2)} \right) \frac{h_e^{\frac{5}{2}}}{p^2\sqrt{120}} + \frac{K_2}{\alpha(1+c_1^2)} \frac{h_e^{\frac{3}{2}}}{p\sqrt{12}} \right] \max \left| n''(\xi) \right|_{\xi \, \epsilon \, \Omega_e} \tag{3.56}$$

or for no-flux problems the estimate (‡) is divided by $\bar{n}_h^e h_e^{1/2}$:

$$REE_e := \frac{1}{\bar{n}_h^e h_e^{1/2}} \left[\left(1 + \frac{K_3}{\lambda} \right) \frac{h_e^{\frac{5}{2}}}{p^2\sqrt{120}} \right] \max \left| n''(\xi) \right|_{\xi \, \epsilon \, \Omega_e} \tag{3.57}$$

According to the inequalities derived in sections 3.6 and 3.7 we would expect that :

$$REN_e \leq REE_e \tag{3.58}$$

for each element e of a given discretization.

In each of the numerical case studies of chapter 2 the numerical solution was obtained using the Galerkin formulation of the problem over a refined discretization of the domain. The REE values were calculated for each element of the discretization. In each case the constants in the estimates (†) and (‡) were approximated as described in section 3.9 and each REE value was plotted as an odot ($\odot$) against its v_b^e value. The REN values were also calculated using the numerical and analytical solution to each problem. These values are shown plotted as boxes ($\square$) against their v_b^e values (see figures 3.1 to 3.4).

3.10.2 An Aggregation Problem

PBE under study

$$\frac{n(v) - n_{in}(v)}{\tau} = \int_0^{v/2} \beta(v-w,w)n(v-w)n(w)dw - n(v)\int_0^{\infty} \beta(v,w)n(w)dw$$

$$\text{where :} \quad n_{in}(v) = \exp(-v), \quad \beta(v,w) = \beta_0 = 1, \quad \tau = 2 \times 10^6$$

Domain considered : $v \in (0, 3.403 \times 10^7)$

Geometric Discretization : 45 elements with first element of length $v_1 = 2$

REE : Calculated using expression (‡)

In figure 3.1 the REE and REN values can be observed to behave as predicted by inequality (3.58) : in each case the REE values bound their corresponding REN values from above. It should also be noted that the errors in the numerical solution are quite small over most of the domain. In the mid-regions of the domain the REE values (which represent predicted upper bounds of the error) are typically around 2%. The REN values (which represent the actual errors) are actually much smaller, typically around 0.1%. These errors increase by as much as two orders of magnitude as the upper and lower limits of the domain are approached.

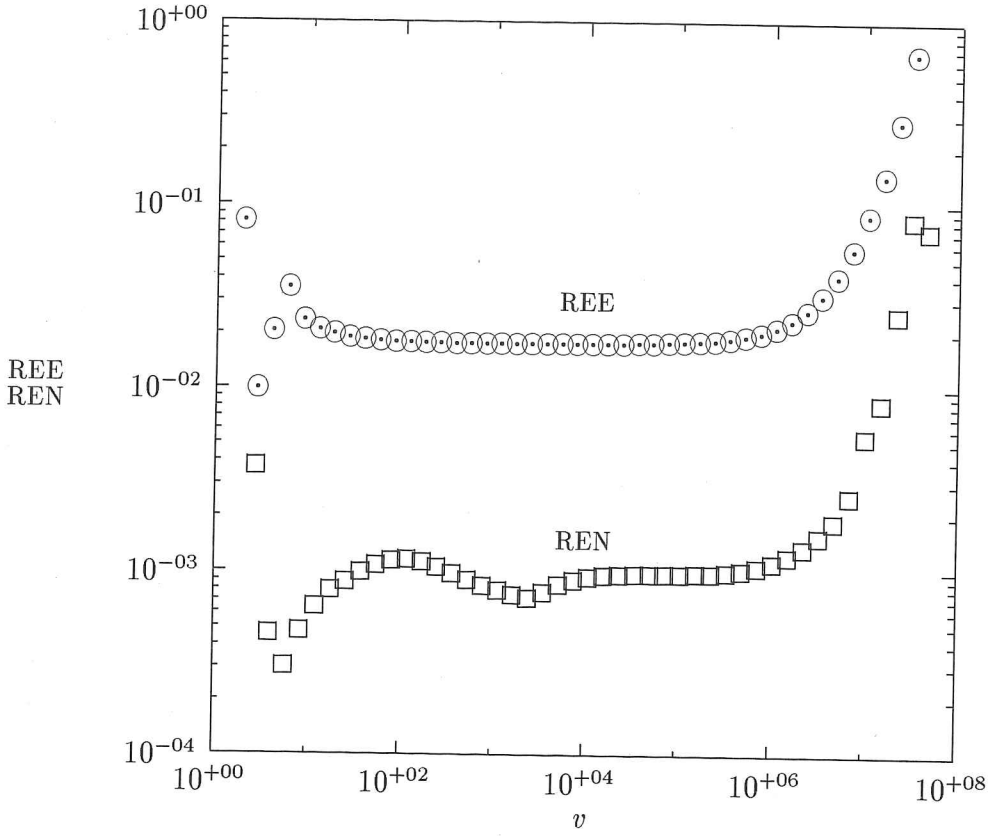

Figure 3.1: REN and REE values for aggregation problem 1c where $n_{in}(v) = \exp(-v)$, $\beta(v,w) = \beta_0 = 1$ and $\tau = 2 \times 10^6$

3.10.3 A Breakage Problem

PBE under study

$$\frac{n(v) - n_{in}(v)}{\tau} = \int_v^\infty \rho(v,w)S(w)n(w)dw - n(v)S(v)$$

$$\text{where :} \quad n_{in}(v) = \exp(-v), \quad \rho(v,w) = \frac{2}{w}, \quad S(v) = v \quad \tau = \times 10^4$$

Domain considered : $v \in (0, 12.981)$

Geometric Discretization : 50 elements with first element of length $v_1 = 10^{-4}$

REE : Calculated using expression (‡)

In figure 3.2 the REE values may again be observed to be upper bounds of their respective REN values. In this case, slightly larger errors exist in the numerical solution. In the mid-region of the domain, the predicted upper bounds of the errors (REE values) are typically around 10% while the actual errors (REN values) are about 1%. Once again these errors dramatically increase in the tail region.

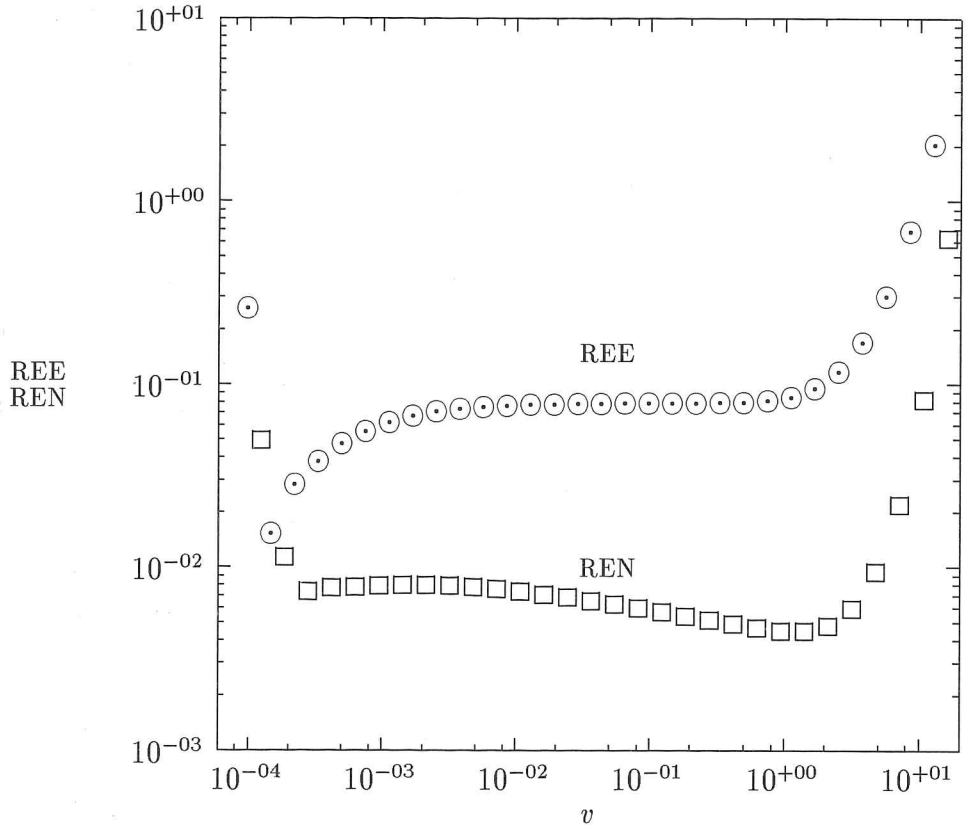

Figure 3.2: REN and REE values for breakage problem 2c with $n_{in}(v) = \exp(-v)$, $\rho(v,w) = 2/w$, $S(v) = v$ and $\tau = 10^4$

3.10.4 A Growth Problem

PBE under study

$$\frac{n(v) - n_{in}(v)}{\tau} + \frac{d}{dv}\Big(G(v)n(v)\Big) = 0$$

where : $n_{in}(v) = v \times \exp(-v)$, $G(v) = G_0 = 1$, $n(0) = 0$ $\tau = 1 \times 10^3$

Domain considered : $v \,\epsilon\, (0, 11440)$

Geometric Discretization : 30 elements with first element of length $v_1 = 0.5$

REE : Calculated using expression (†)

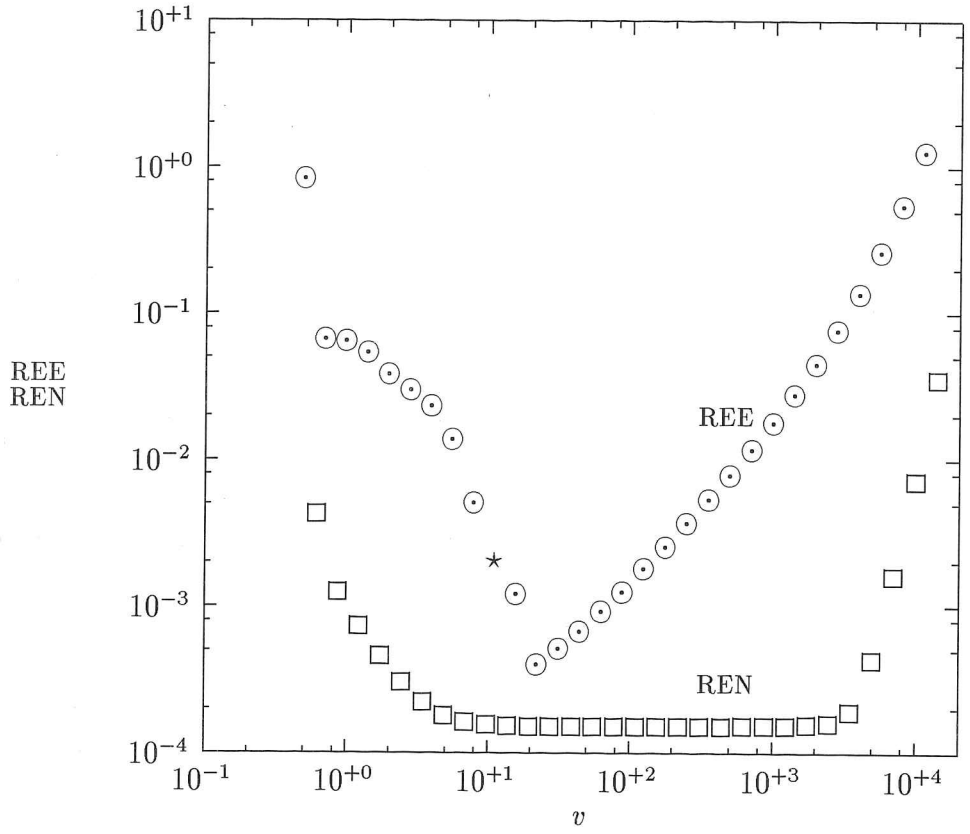

Figure 3.3: REN and REE values for growth problem 3c where $n_{in}(v) = v\exp(-v)$, $G(v) = G_0 = 1$, $n(0) = 1$ and $\tau = 10^3$. The star denotes an element where the error estimate becomes singular. This singularity will be discussed in subsection 3.11.2.

In figure 3.3 the REE values may once again be observed to bound their respective REN values from above. In this case however the REE values do not predict the correct trend in the behaviour of the REN values. The REN values are reasonably uniform in the mid-region of the domain whereas the REE values decrease to a minimum and then continually increase with increasing volume. This issue will be discussed further in subsection 3.11.4.

The star ($\star$) appearing in 3.3 denotes an element where the error estimate becomes singular. The identification of this singularity will be provided in subsection 3.11.2

3.10.5 A Problem of Combined Aggregation, Growth and Nucleation

PBE under study

$$\frac{n(v) - n_{in}(v)}{\tau} + \frac{d}{dv}\Big(G(v)n(v)\Big) = \int_0^{v/2} \beta(v-w,w)n(v-w)n(w)dw - n(v)\int_0^{\infty} \beta(v,w)n(w)dw \quad (3.59)$$

where : $n_{in}(v) = \exp(-v)$, $\beta(v,w) = \beta_0 = 100$, $G(v) = G_0 = 100$, $\tau = 1$

Domain considered : $v \in (0, 2.365 \times 10^5)$

Geometric Discretization : 40 elements with first element of length $v_1 = 1$

REE : Calculated using expression (†)

Once again in figure 3.4 it can be seen that the REE values bound their respective REN values from above. As in the aggregation and breakage problems the REE values also predict the correct trend in the behaviour of the REN values. Reasonably uniform errors occur in the mid-region of the domain (about 1.5% and 0.8% for the REE and REN values respectively). These errors increase by nearly two orders of magnitude as the tail region is reached.

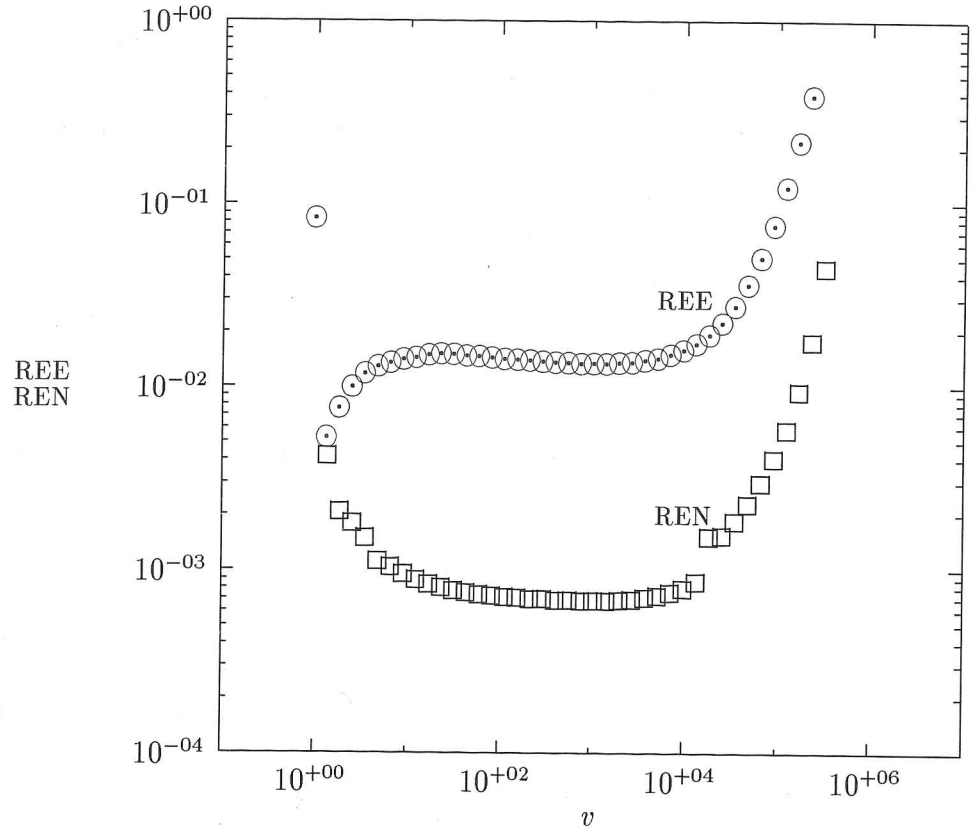

Figure 3.4: REN and REE values for the aggregation, growth and nucleation problem 4c where $n_{in}(v) = \exp(-v)$, $\beta(v,w) = \beta_0 = 100$, $G(v) = G_0 = 100$ and $\tau = 1$

3.11 Discussion

The derived estimates (†) and (‡) permit upper bounds to be calculated for the errors in finite element solutions to problems of the form (3.1) or (3.24). This calculation may be performed without any knowledge of the analytical solution.

3.11.1 The Numerical Case Studies

In figures (3.1, 3.2, 3.3 and 3.4) it can be seen without exception that the calculated REE values are greater than the REN values. This result is as predicted by inequality (3.58).

In most cases the REE values over-estimate the actual errors by an order of magnitude. This is still a reasonably tight bound considering that the analytical solution to the problem can vary by as much as sixteen orders of magnitude over the domain of interest (see figures 2.6, 2.7, 2.9 and 2.10).

Additional sources of error are present in the obtained finite element solutions due to : numerical integration, finite floating point arithmetic and truncation of the infinite domain to a finite upper limit. The error estimate derived in this chapter does not take any of these effects into account.

The error estimate also assumes that the finite element formulation of the problem is solved exactly. This is not the case when aggregation terms are present since the resulting system of equations is non-linear and can only be approximately solved with a finite number of iterations.

3.11.2 Singularity of the Error Estimate

Inherent in the derivation of the error estimates are assumptions **(A2)** and **(A5)** which require q_2 to be bounded from above. Expressions (3.45) and (3.46) reveal that q_2 may become singular where $n'(v) = 0$. Hence this assumption is only valid in elements where the solution is monotonically increasing or decreasing.

In the growth problem a singularity in the error bound occurs at the mode of the solution. A finite REE value was obtained for this element since it is assumed that q_2 is either monotonically increasing or decreasing within each element hence ~~it's~~ HS value

was tested for a maximum only at the end-points of this element and not at the mode. In figure (3.3) the element containing the mode has been tagged by a star ($\star$) to denote that the value plotted is not strictly valid.

3.11.3 Convergence of the Error Estimate

The error estimate may also be used to establish convergence of the finite element solution to the analytical solution for problems of the forms (3.1) and (3.24).

Consider the error estimate applied to a problem in which the domain is increasingly refined towards an infinite number of elements of infinitesimally small lengths, then :

$$\lim_{h_e \to 0} \left\| u - u_h \right\|_{\Omega_e} \leq \lim_{h_e \to 0} \left[\left(1 + \frac{K_1}{\alpha(1+c_1^2)} \right) \frac{h_e^{\frac{5}{2}}}{p^2\sqrt{120}} + \frac{K_2}{\alpha(1+c_1^2)} \frac{h_e^{\frac{3}{2}}}{p\sqrt{12}} \right] = 0 \quad (3.60)$$

If the above inequality is to be applied to the steady state PBE an additional limit $v_{max} \to \infty$ must also be considered.

The right hand limit of inequality (3.60) will tend towards zero provided that :

1. All of the declared assumptions are valid.

2. Neither of the terms $\frac{K_1}{\alpha(1+c_1^2)}$ or $\frac{K_2}{\alpha(1+c_1^2)}$ become infinitely large as the element lengths are decreased. Or more precisely, for REE values (and consequently REN values) to converge to zero then h_e^2 must tend to zero faster than $\frac{K_1}{\alpha(1+c_1^2)}$ tends to ∞ and h_e must tend to zero faster than $\frac{K_2}{\alpha(1+c_1^2)}$ tends to ∞.

The behaviour of these two terms was investigated as the element length was decreased for the combined aggregation, growth and nucleation problem (case 4a). Each of the terms was plotted against the volume coordinate of its upper limit (v_b^e) over domains geometrically partitioned into 20, 40 and 80 elements. These results are shown in figures 3.5 and 3.6.

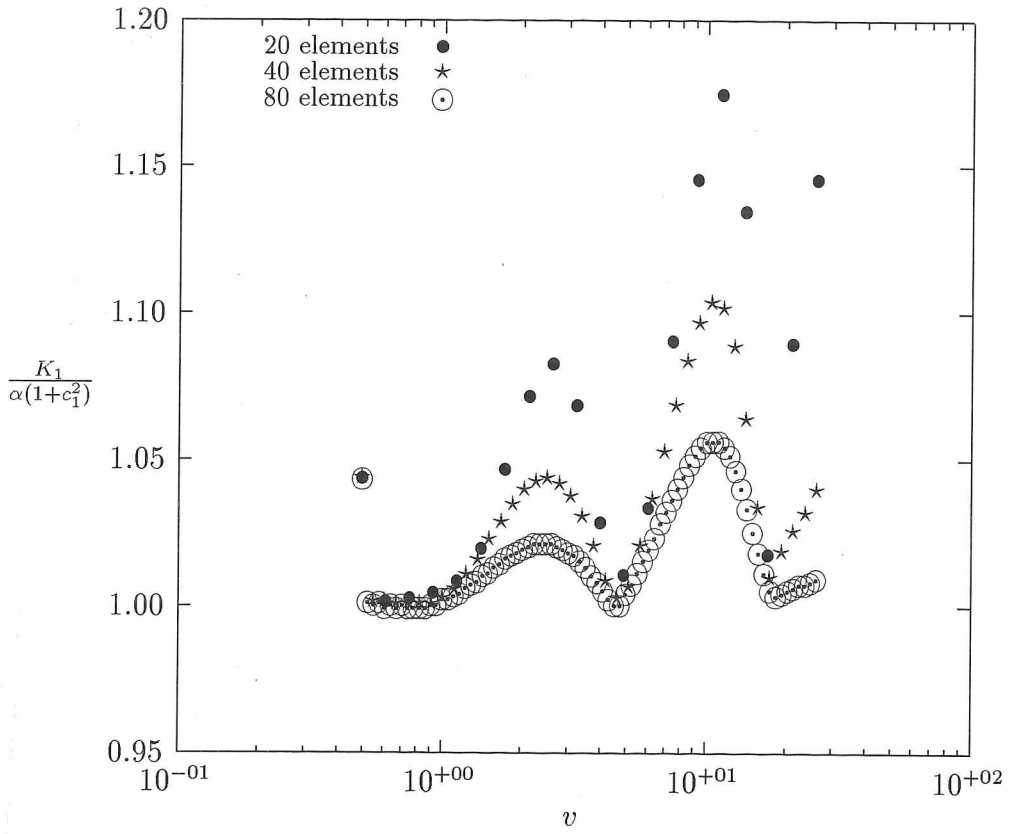

Figure 3.5: Numerical values of $\frac{K_1}{\alpha(1+c_1^e)}$ versus v_b^e for problem 4c with domains geometrically partitioned into 20, 40 and 80 elements

In figure 3.5 the $\frac{K_1}{\alpha(1+c_1^e)}$ terms can be seen to decrease towards unity as the element lengths become smaller. While in figure 3.6 the $\frac{K_2}{\alpha(1+c_1^e)}$ terms asymptote towards a single curve as the element lengths become smaller. This curve appears to be independent of the discretization. Similar observations were made for the other cases of the PBE. Hence the constants are sufficiently well behaved for the limit (3.60) to hold true and we can conclude that in cases where the declared assumptions are valid, the error estimate establishes convergence of the finite element solution to the analytical solution.

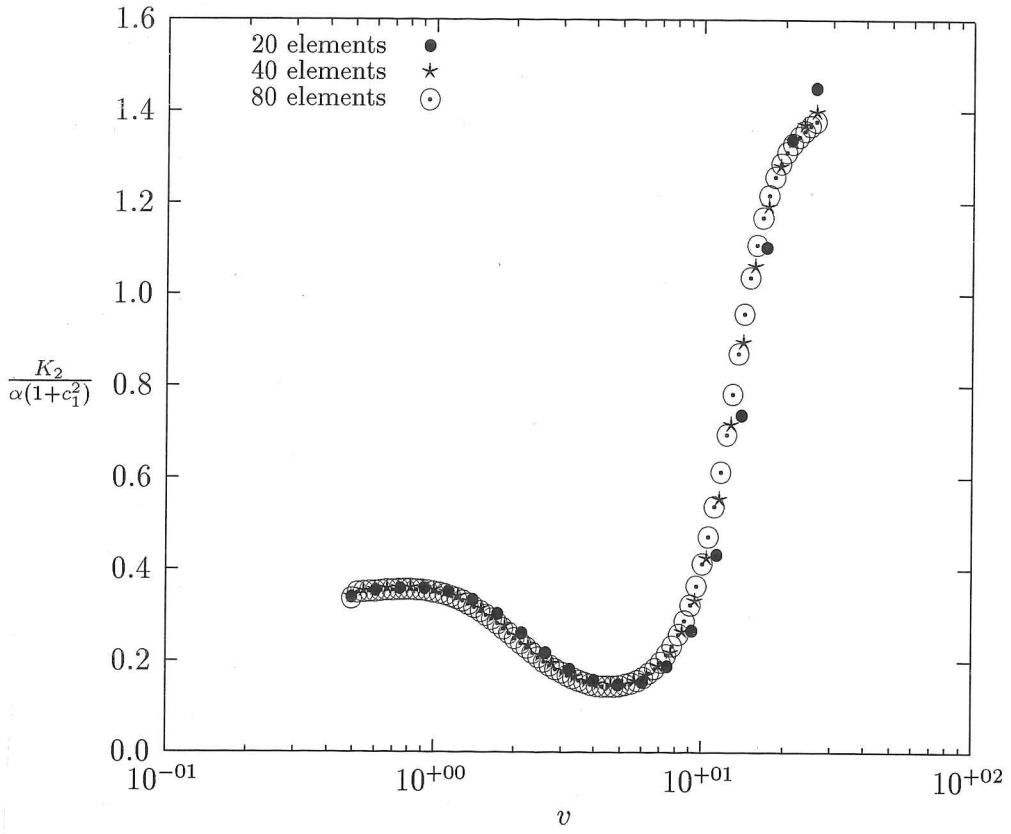

Figure 3.6: Numerical values of $\frac{K_2}{\alpha(1+c_1^e)}$ versus v_b^e for problem 4c with domains geometrically partitioned into 20, 40 and 80 elements

3.11.4 The Error Estimate and Automatic Refinement

The REN values of figures (3.1, 3.2, 3.3 and 3.4) were obtained over meshes refined according to a geometric progression. In each case the length of the first element was reduced and the zeroth moment calculated until further reductions yielded insignificant changes (less than 0.5%) in the zeroth moment. Then the number of elements was increased until further increases yielded insignificant changes (less than 0.5%) in the second moment. This type of refinement results in solutions with regions of very high accuracy (REN values are as low as 10^{-3} in the mid-regions of figure 3.1) and other regions where very poor accuracy is achieved (REN values as high as 10^{-1} in the tail-region of figure 3.1). If an "optimal" mesh may be defined as one in which

the REN values are equally distributed then meshes refined according to geometric schemes are clearly sub-optimal. Also of concern is the fact that large errors (REN $> 10^{-1}$) can be present but not detected using a criterion based on the second moment to terminate refinement.

In figures (3.1, 3.2 and 3.4) it can be seen that REE values predict correctly the trends in REN values ie. high/low REE values correspond to high/low REN values. Hence the error estimates may be used to locate regions where the element lengths should be reduced or increased if an optimal mesh is to be obtained.

In the growth problem the REE values did not predict the trends in the REN values. In figure 3.3 the REE values can be seen to be increasing in a region where the REN values are relatively constant. This poor correlation results from the fact that error estimate (†) was derived for a nonlinear equation whereas the growth only problem is a linear equation.

Since the error estimates are derived in terms of the elements lengths h_e they not only indicate where refinement is needed but also imply how much each element length should be increased or decreased if an optimal mesh is to be obtained. This information will form the basis of the automatic mesh refinement algorithm to be presented in the next chapter.

3.12 Chapter Conclusions

- In all investigated cases the error estimates (†) and (‡) reliably predicted upper bounds to the actual error present in obtained numerical solutions. This was achieved without any knowledge of the analytical solution.
- These error estimates also establish convergence of finite element solutions to analytical solutions of problems of the forms (3.1) and (3.24).

- These error estimates make it possible to develop a rigorous approach to automatic mesh refinement for the steady state PBE

Chapter 4

Automatic Mesh Refinement

Numerical methods currently used for solving the steady state population balance equation employ heuristic procedures to discretize and refine domains prior to solving a problem. Such heuristics require much guess-work to implement and typically result in solutions that are of poor quality in some regions and over-specified in others.

In this chapter a refinement algorithm is proposed whereby the user specifies a maximum permissible error and the algorithm automatically finds a new discretization such that this error is not exceeded in any region of the domain.

The automatic refinement algorithm was applied to several cases of the steady state population balance equation and was found to result in more accurate solutions and to be computationally more efficient than refinement according to heuristics.

4.1 The Importance of Discretization

The majority of numerical techniques that have been used to solve the steady state PBE may be described as "discrete methods". As such they require the domain to be sub-divided into smaller domains prior to solving the PBE. All discrete methods are guaranteed to perform "poorly" if they are supplied with "poor" discretizations. A discretization that is too sparse will result in a poor approximation of the problem and consequently a poor numerical solution will be obtained. Errors of about 100 % can be observed in figure 3.2 due to a discretization that is too sparse in the tail region. A discretization that is too fine can greatly increase the number of unknowns in the system without significantly improving the quality of the obtained solution. For instance if case study 1c was to be solved over a uniformally discretized domain to an accuracy $REN \approx 0.5\%$ it would need to be spanned by elements of length $h_e \approx 2\ units$ thus 17 015 000 elements would be required to span the domain. This corresponds to a system of 51 045 001 unknowns. A similar order of accuracy will be achieved for this problem in section 4.7.2 using a "more optimal" discretization consisting of 32 elements that correspond to just 97 unknowns.

In the case of the PBE an "optimal" discretization is very difficult to acquire since the solution to the problem varies by many orders of magnitude over the domain of interest. Consequently "optimal" discretizations can in many cases consist of elements of lengths that change by several orders of magnitude over the domain of interest.

In short, great care must be taken with domain discretization if accurate solutions are to be obtained using reasonable amounts of computer power.

4.2 Previous Approaches to Discretization and Refinement

A rigorous approach to domain discretization and refinement is yet to be proposed for the steady state PBE. The majority of methods use heuristic procedures to subdivide the domain into an initially coarse discretization. The two most popularly used heuristics are discretization according to a geometric progression (as used by Hounslow (1990), Gelbard, Tambour and Seinfeld (1980), Hill and Ng (1995) and Hill and Ng (1996)) and uniform discretization whereby the domain is spanned by sub-domains of equal length (as used by Gelbard and Seinfeld (1978), Steemson and White (1988) and Pilinis (1990)). An initially poor solution is then improved upon by re-solving the problem over a finer discretization. This process is usually termed "refinement."

Several problems are inevitably encountered when attempting to refine a discretization according to any such heuristics :

1. Too few adjustable parameters are available to obtain an "optimal" discretization. In the case of geometric discretization only the length of the first element and the number of elements may be adjusted while refinement according to a uniform discretization is even more restrictive : only the number of elements may be adjusted. It is easy to foresee that in many cases refinement according to these schemes will be either excessively expensive (in a computational sense) or yield poor solutions (two such examples have been mentioned in the previous section).

2. It is difficult to determine when to terminate the refinement procedure ie. at what stage further refinement will yield negligible improvements in the obtained solution. To this end, the behaviour of a functional (for example the second moment) is observed and the refinement procedure is terminated upon its con-

vergence. However it has been demonstrated previously (see section 3.10) that it is possible for such functionals to converge while large errors in the numerical solution are still present in some regions.

3. Considerable guesswork is required when progressing from one refinement stage to the next. For instance even when observing the behaviour of a converging functional it is difficult to determine how many additional elements should be added before the problem is re-solved again.

Two attempts have been made to address the above-mentioned issues : Eyre, Wright and Reuter (1988) proposed a refinement procedure in which the arc length of the solution within each sub-domain was equilibrated at each refinement stage. While Kumar and Ramkrishna (1996a) suggest that the length of each sub-domain could be adjusted proportionally to the change in magnitude of the solution within it.

Both these procedures automate the refinement procedure. No decision needs to be made by the user concerning how many additional sub-domains should be used or when the refinement procedure should be terminated. Another desirable feature of these methods is that each sub-domain of the discretization effectively becomes an adjustable parameter.

Unfortunately neither of these refinement procedures can be deemed as being successful since that of Eyre, Wright and Reuter degraded the quality of the final obtained solution, while the approach suggested by Kumar and Ramkrishna was investigated in a finite element method by Nicmanis (1995) (it was never actually implemented by Kumar and Ramkrishna) and it was found in many cases that refinement according to a geometric progression would be computationally more efficient.

The failure of the above-mentioned approaches to automatic refinement can be mainly attributed to the fact that they are based upon local considerations. A more sophisticated approach is necessary due to the presence of the integral terms of the PBE. Consider for instance the death term due to aggregation. This term is a functional

of the solution over the *entire* domain. Refinement and hence subsequent improvement of the solution in one region of the domain therefore renders improvements in the quality of the solution in every other region of the domain. Similar albeit more complicated effects also occur due to the other integral terms.

4.3 A New Approach to Discretization

In this section a new approach to domain discretization and refinement will be proposed. The new refinement algorithm allows the user to specify a maximum acceptable error tolerance and a new discretization is found such that this error tolerance is not exceeded at any point of the domain.

The key to this refinement method is the error estimate derived in the previous chapter. These estimates predict the error for a given discretization. The approach taken is to reverse this procedure. The constants of expressions (†) and (‡) are obtained using a cheaply obtained numerical solution over a coarse discretization. Once a maximum permissible error is specified the error estimate may be used in reverse to solve for a new, more optimal discretization.

This approach embodies the desirable characteristic of being automatic. The decision as to "how many new sub-domains should be added" is automatically dictated by the error estimate which determines the number of sub-domains required to reduce the error below a pre-specified bound. Termination of the refinement procedure is also automatic : it ends when the error in each sub-domain is reduced below its pre-specified upper tolerance. A new discretization is constructed at each refinement step hence the length of each sub-domain may be considered as an adjustable parameter.

Unlike the earlier endeavours of Eyre, Wright and Reuter (1988) and Kumar and Ramkrishna (1996a), the new procedure is not based upon local considerations. The global behaviour of the solution is accounted for by assumptions **(A2)**, **(A3)** and

(A5) which are upper bounds of partial derivatives of an operator with respect to the solution over the *entire* domain. This feature will permit the algorithm to be used in aggregation and breakage problems.

4.4 Definition of an "Optimal Discretization"

For the purposes of this chapter a discretization will be defined as optimal if the error in the solution derived over this discretization is equally distributed over each of the sub-domains comprising it.

This definition requires the specification of an error quantity to be equilibrated. Ideally the REN values would be equilibrated however they cannot be determined without the analytical solution. Their behaviour can however be inferred by the REE values (which can be determined). In the simulations of section 3.10 it was shown that the REE correctly predicts trends in the REN. Hence the criterion used for optimizing a discretization will be to equilibrate the REE which should also equilibrate the REN.

This methodology automates the refinement procedure as well as its termination whilst permitting the length of each sub-domain to be used as an adjustable parameter but unlike previous heuristic approaches of Eyre, Wright and Reuter (1988) and Kumar and Ramkrishna (1996a), it has a sound theoretical basis.

4.5 The Automatic Refinement Algorithm

An overview of the refinement procedure is given here and is followed by a more detailed discussion of steps 3 and 4 :

1. Specify the maximum relative error tolerance that is permissible in each element. This quantity will be denoted as (Δ).

2. Obtain a solution over a coarse discretization.

3. Calculate the constants in the error estimate for each element in this coarse discretization.

4. Set Δ = REN and solve for the element lengths that satisfy this relation in each region of the discretization.

In the above procedure the constants in the error estimate (†) are calculated for each element. An example of this step was performed in the previous chapter and the grouped constants $\frac{K_1}{\alpha(1+c_1^2)}$ and $\frac{K_2}{\alpha(1+c_1^2)}$ were plotted for each element as points against their respective v_b values in figures 3.5 and 3.6. During the refinement procedure the values of each of the constants are assumed to be uniform within the domain of each element. Hence for the purposes of the refinement procedure figures 3.5 and 3.6 would be more descriptively represented as step functions with uniform values within each element.

Step 4 of the above procedure is performed as follows :

Say for instance that an appropriate element length (h_{new}) is to be determined in the vicinity of a point which is contained in element e such that the error tolerance Δ is not exceeded. Error estimate (†) is used to provide a relationship between the error within an element and the length of the element. First we set Δ =REE :

$$\Delta = \kappa_1^e h_{new}^2 + \kappa_2^e h_{new} \tag{4.1}$$

where the following simplifying notation has been introduced :

$$\kappa_1^e := \frac{1}{n_h^e}\left(1 + \frac{K_1^e}{\alpha_e(1+[c_1^e]^2)}\right)\frac{\max|u''(\xi)|_{\xi\,\epsilon\,\Omega_e}}{p^2\sqrt{120}} \tag{4.2}$$

and :

$$\kappa_2^e := \frac{1}{n_h^e}\left(\frac{K_2^e}{\alpha_e(1+[c_1^e]^2)}\right)\frac{\max|u''(\xi)|_{\xi\,\epsilon\,\Omega_e}}{p\sqrt{12}} \tag{4.3}$$

we then solve the quadratic equation :

$$0 = \kappa_1^e h_{new}^2 + \kappa_2^e h_{new} - \Delta \tag{4.4}$$

all constants in the above expression are positive so if we permit only positive element lengths the solution to this equation is :

$$h_{new} = \frac{-\kappa_2^e + \sqrt{(\kappa_2^e)^2 + 4\kappa_1^e \Delta}}{2\kappa_1^e} \tag{4.5}$$

Calculations of this nature are used along the domain to generate a new discretization as follows :

1. Calculate the length of the first element using the relation :

$$h_1 = \frac{-\kappa_2^1 + \sqrt{(\kappa_2^1)^2 + 4\kappa_1^1 \Delta}}{2\kappa_1^1} \tag{4.6}$$

2. The length of each subsequent i^{th} element is calculated using the relation :

$$h_i = \frac{-\kappa_2^e + \sqrt{(\kappa_2^e)^2 + 4\kappa_1^e \Delta}}{2\kappa_1^e} \tag{4.7}$$

   where e is the element containing the volume coordinate $v = \sum_{j=1}^{i-1} h_j$

3. The discretization procedure is completed (for this refinement stage) when a volume coordinate is obtained such that :

$$v = \sum_{j=1}^{i} h_j > v_{\max} \tag{4.8}$$

   in which case the final element length is set to :

$$h_i = v_{\max} - \sum_{j=1}^{i-1} h_j \tag{4.9}$$

Nodal values of the solution over the refined domain can be obtained by interpolation of the solution over the previous discretization. These values can be used to initialize the iterative solution procedure so that the problem may be re-solved more quickly over the newly refined domain.

Upon converging a solution over the newly refined domain new REE values are calculated for each element and if any of them significantly exceed Δ the refinement procedure is repeated.

4.6 A Simplified Version for No-Flux Problems

In cases of the PBE where no particle growth occurs ($G(v) = 0$) the estimate (‡) is used to provide a relationship between the error contained within an element and the length of the element. To determine an appropriate element length (h_{new}) in the vicinity of a point contained within element e such that a REE value of Δ is obtained, we set $\Delta = \text{REE}$:

$$\Delta = \kappa_3^e h_{new}^2 \tag{4.10}$$

where :

$$\kappa_3^e = \frac{1}{n_h^e}\left(1 + \frac{K_3^e}{\Lambda_e}\right)\frac{\max|n''(\xi)|_{\xi\,\epsilon\,\Omega_e}}{p^2\sqrt{120}} \tag{4.11}$$

we then solve for the appropriate element length :

$$h_{new} = \sqrt{\frac{\Delta}{\kappa_3^e}} \tag{4.12}$$

These relationships are used to discretize the domain as previously described in section 4.5.

4.7 Numerical Case Studies

In this section numerical case studies are performed for each of the problems investigated in section 2.14. The automatic refinement procedure is applied to case a of each problem and any improvements in the overall quality of the solution, the additional cost of implementing the procedure and characteristics of the initial and refined discretizations are recorded.

These specific examples will then be followed by series of simulations where higher quality solutions are sought by increasing the number of elements composing both geometrically and automatically refined domains. This is done to ensure that the above examples are not merely isolated cases where the refinement procedure improves the performance of the finite element algorithm. Records are made of improvements in the solution and the CPU requirements as the discretization becomes finer over both geometrically and automatically refined domains.

4.7.1 Assessment of the Quality of a Solution

At this stage the following quantity is introduced as a quantitative measure of the *overall quality* of an obtained numerical solution :

$$rL_2 = \sqrt{\frac{1}{v_{max}} \int_0^{v_{max}} \left(\frac{n - n_h}{n} \right)^2 dv} \tag{4.13}$$

This quantity is effectively a relative error in the numerical solution averaged over the *entire domain.*

4.7.2 An Aggregation Problem

The automatic mesh refinement procedure was applied to the steady state PBE for aggregation (case 1a of chapter 2). The parameters of this problem are summarized below :

PBE under study

$$\frac{n(v) - n_{in}(v)}{\tau} = \int_0^{v/2} \beta(v-w, w) n(v-w) n(w) dw - n(v) \int_0^{\infty} \beta(v, w) n(w) dw$$

$$\text{where :} \quad n_{in}(v) = \exp(-v), \quad \beta(v, w) = \beta_0 = 1, \quad \tau = 200$$

Domain considered : $v \in (0, 3306)$

These specific examples will then be followed by series of simulations where higher quality solutions are sought by increasing the number of elements composing both geometrically and automatically refined domains. This is done to ensure that the above examples are not merely isolated cases where the refinement procedure improves the performance of the finite element algorithm. Records are made of improvements in the solution and the CPU requirements as the discretization becomes finer over both geometrically and automatically refined domains.

4.7.1 Assessment of the Quality of a Solution

At this stage the following quantity is introduced as a quantitative measure of the *overall quality* of an obtained numerical solution :

$$rL_2 = \sqrt{\frac{1}{v_{max}} \int_0^{v_{max}} \left(\frac{n - n_h}{n} \right)^2 dv} \tag{4.13}$$

This quantity is effectively a relative error in the numerical solution averaged over the *entire domain.*

4.7.2 An Aggregation Problem

The automatic mesh refinement procedure was applied to the steady state PBE for aggregation (case 1a of chapter 2). The parameters of this problem are summarized below :

PBE under study

$$\frac{n(v) - n_{in}(v)}{\tau} = \int_0^{v/2} \beta(v-w, w) n(v-w) n(w) dw - n(v) \int_0^{\infty} \beta(v, w) n(w) dw$$

$$\text{where :} \quad n_{in}(v) = \exp(-v), \quad \beta(v, w) = \beta_0 = 1, \quad \tau = 200$$

Domain considered : $v \in (0, 3306)$

Initially this problem was solved over a geometrically discretized and refined domain :

Summary of Results over a Geometric Discretization :

Number of elements used in the discretization = 32

Length of first element = 1 unit

Number of iterations required for convergence = 262

Resulting rL_2 value = 0.943

The automatic refinement procedure was then applied to the problem over the above mentioned geometric discretization. The Δ value was altered until a final discretization of 32 elements was obtained. This was done to ascertain whether (and at what additional cost) the automatic refinement procedure could be used to improve the overall quality of a solution over a geometric discretization.

Summary of Results over an Automatically Refined Discretization :

Δ value = 0.021

Expression used to calculate new element lengths : (4.12)

Number of elements in the final discretization = 32

Number of applications of the refinement procedure = 2

Additional number of iterations required = 77

Final rL_2 value over the automatically refined domain = 0.660

Figures 4.1-4.3 show the REE values for each element plotted against their respective v_b coordinates at each stage of the refinement procedure. The objective of the refinement procedure is to reduce each REE value to below the horizontal line REE = 0.0231 ie. so that no REE value exceeds the tolerance Δ = 0.021 by more than 10 %. This condition was achieved after two applications of the automatic refinement procedure as shown in figure 4.3.

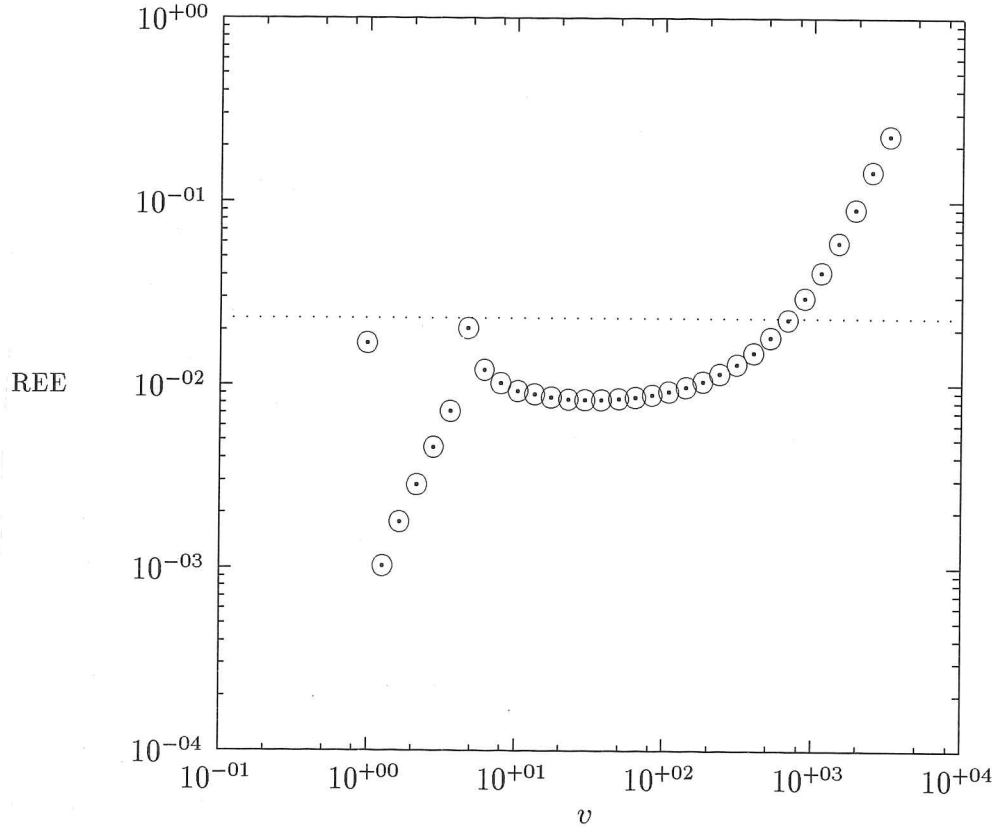

Figure 4.1: REE values for aggregation problem 1a over a geometrically discretized domain

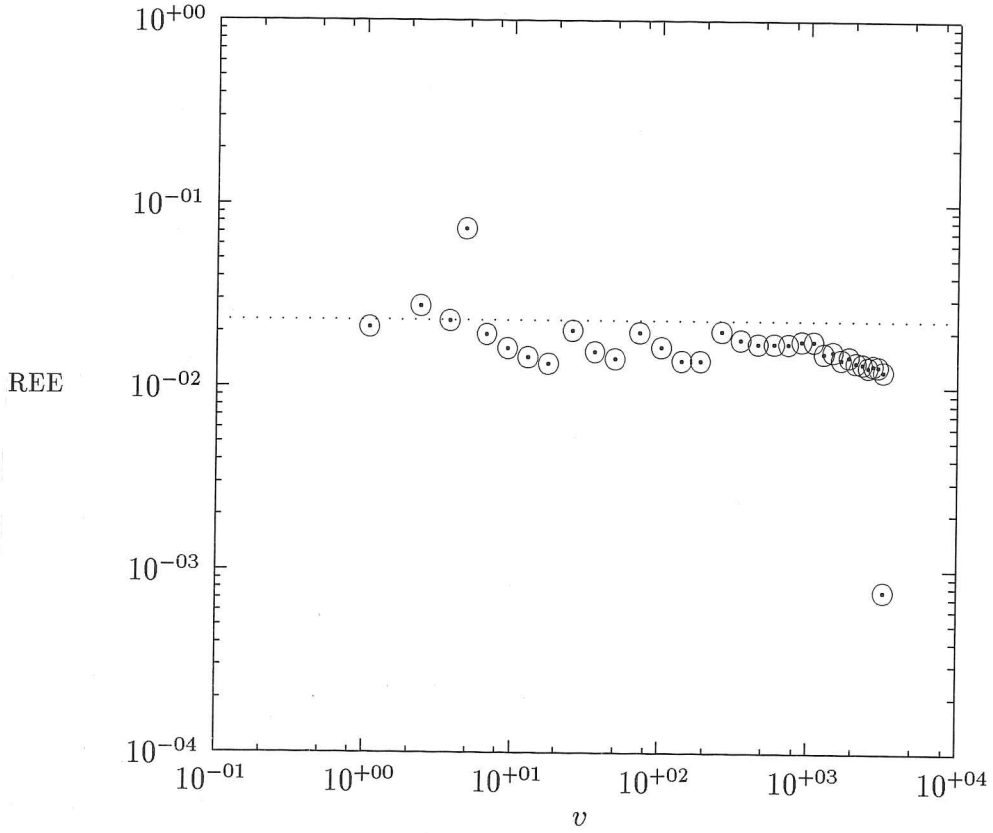

Figure 4.2: REE values for aggregation problem 1a after one application of the automatic refinement procedure

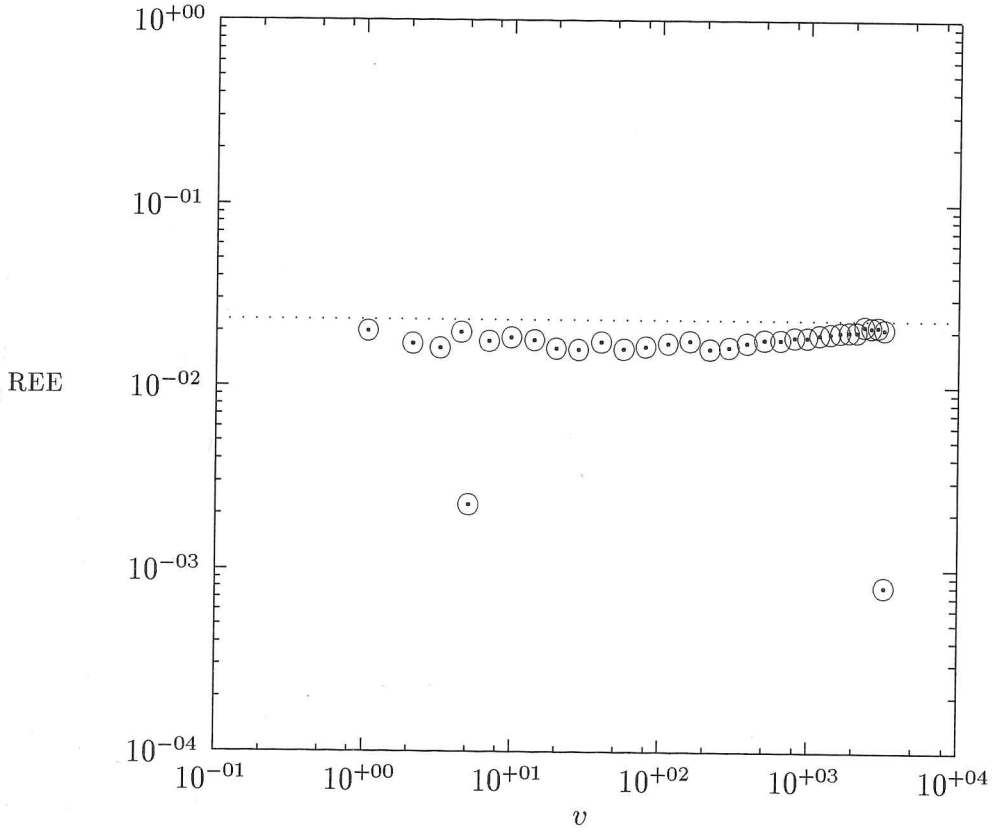

Figure 4.3: REE values for aggregation problem 1a upon completion of the automatic refinement procedure

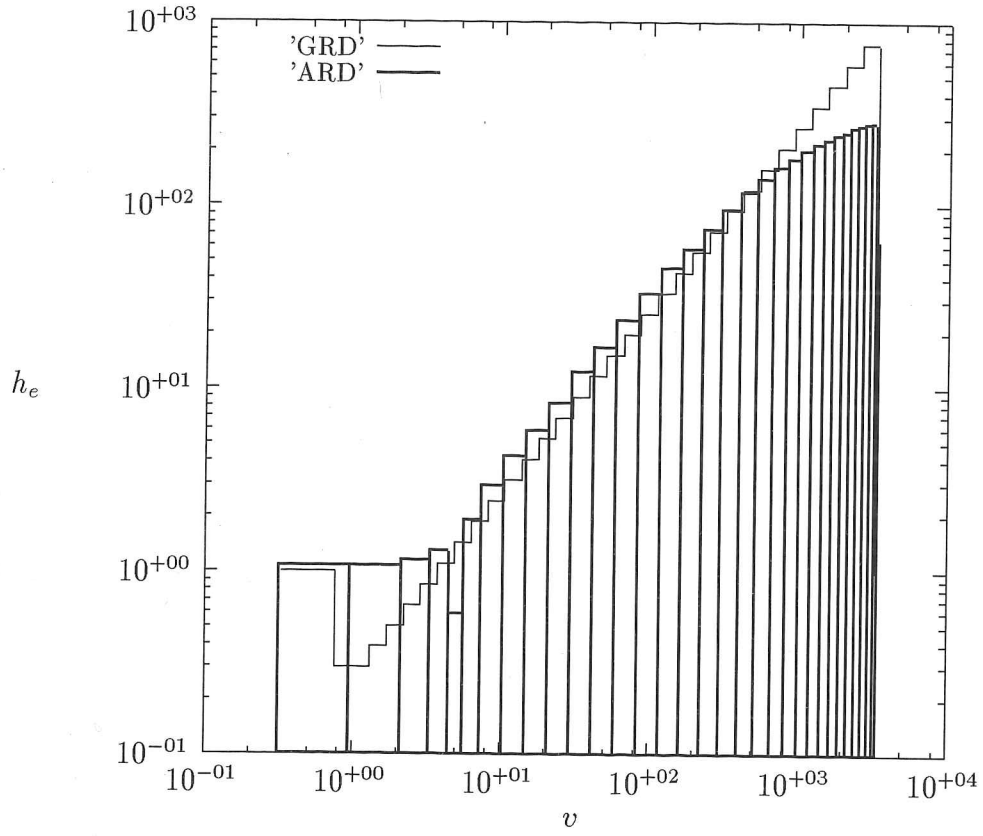

Figure 4.4: Element lengths over a geometrically refined discretization (GRD) and over an automatically refined discretization (ARD) for aggregation problem 1a

A plot was made of the length of each element at their respective volume coordinates. Those of the geometrically refined discretization (GRD) are shown in figure 4.4 as a step function while the element lengths of the automatically refined discretization (ARD) are shown as boxes. The elements comprising the tail region of the ARD have been reduced in length by about a factor of three from their original lengths in the GRD, while in the rest of the domain the ARD is spanned by larger elements than the GRD.

Although the automatic refinement procedure is based upon equilibrating the REE values for each element its ultimate objective is to obtain a more even distribution in the REN values of each element. To observe whether this has been achieved the REN values are plotted in figure 4.5 against their respective volume coordinates for both the geometrically refined discretization ($\star$) and the automatically refined discretization ($\odot$).

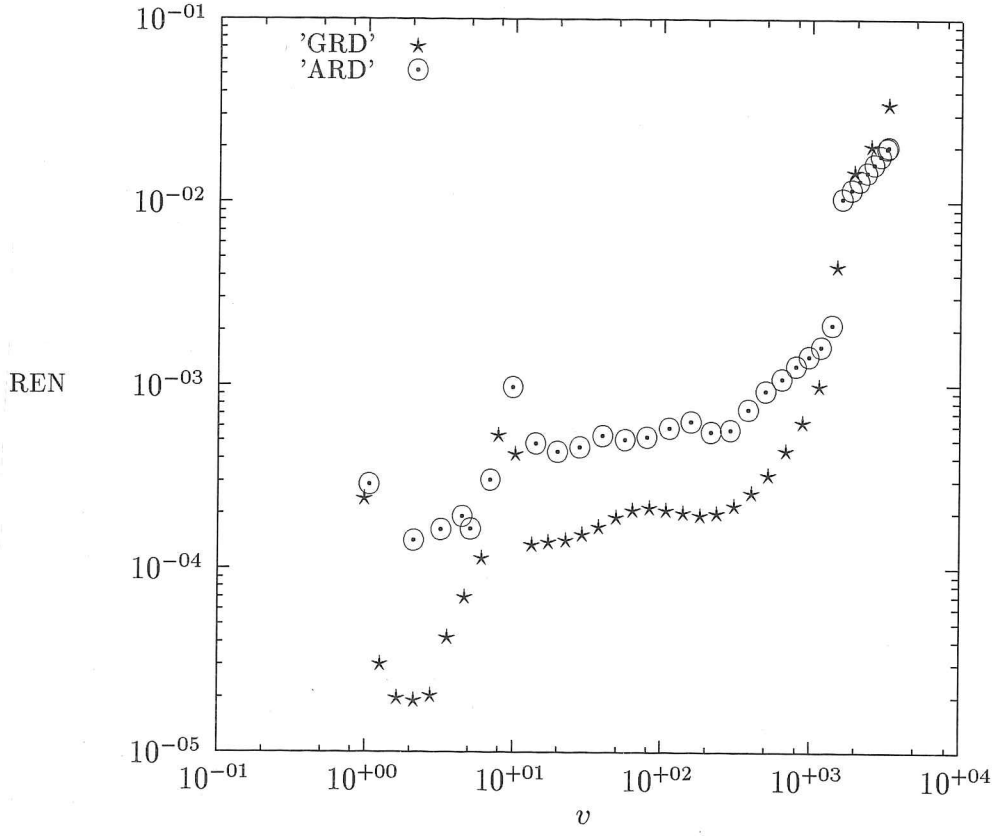

Figure 4.5: REN values over a geometrically refined discretization and over the automatically refined discretization for aggregation problem 1a

The REN values of the ARD are not as uniform as the REE values of figure 4.3. In fact they appear to be only slightly more uniform than the REN values of the GRD (the stars in figure 4.5). From this plot, no definitive conclusions can be made on whether the automatic refinement algorithm improves or degrades the quality of

a numerical solution. This issue will be quantitatively addressed by the series of simulations performed at the end of this subsection.

In figure 4.5 a sharp increase occurs in the REN values of with volume coordinates greater than 1500 units. This is the region where the asymptotic approximation (2.75) is used as the solution to the PBE.

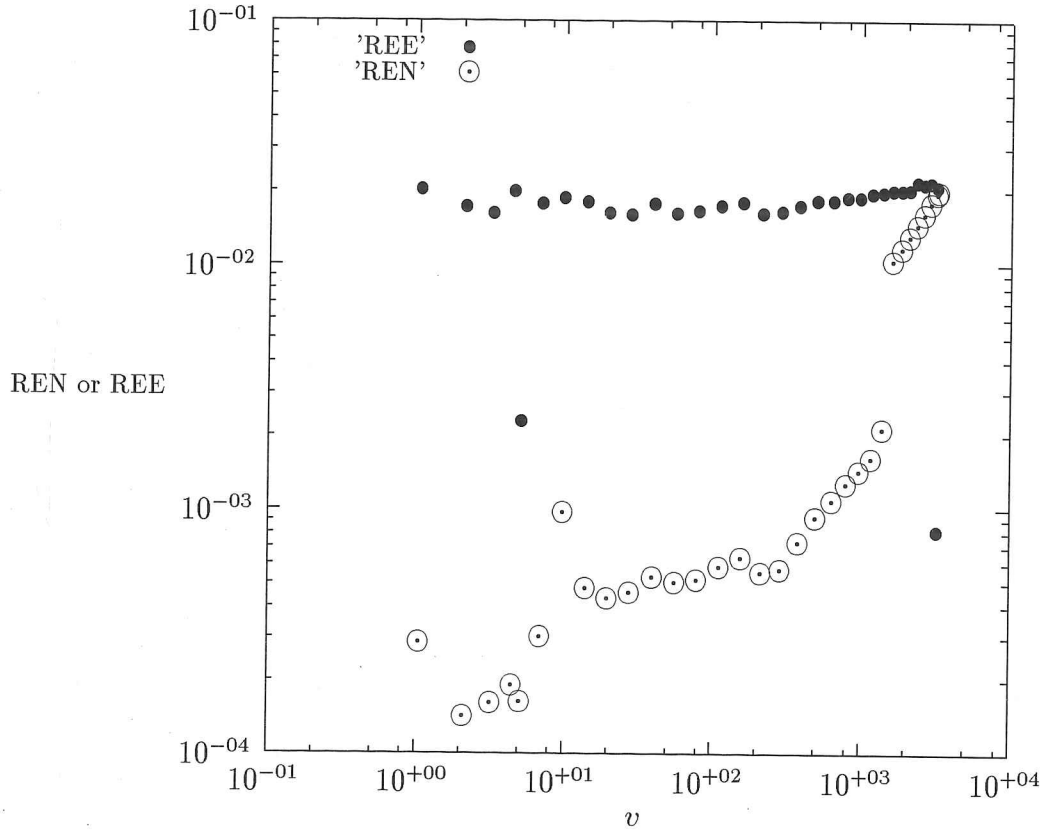

Figure 4.6: REN and REE values over the automatically refined discretization for aggregation problem 1a

The REE and REN values over the automatically refined discretization were calculated to ensure that the refinement procedure did not degrade the ability of the estimate (‡) to predict an upper bound to the error in an obtained numerical solution. This is confirmed in figure 4.6.

Two series of simulations were performed to investigate the effects of geometric refinement and automatic refinement on the rL_2 error and CPU requirements for the aggregation problem. The aggregation problem was solved for a time constant $\tau = 10$ over the domain $v \in (0, 170)$. Over this smaller domain there is no need to use the asymptotic approximation of the solution which was seen in figure 4.5 to significantly affect the REN values.

In the first series of simulations the PBE for aggregation was solved over a domain that was discretized and refined according to a geometric discretization. The initial discretization consisted of ten elements with a first element of length $v_1 = 1.5$ units. The number of elements was increased at each refinement stage (according to the same geometric progression) by increments of five elements until a discretization of 60 elements was obtained. The rL_2 value and CPU requirements for each discretization were recorded.

In the second series of simulations the same problem was solved but the automatic procedure was used to discretize and refine the domain. In each case a solution was obtained over a coarse discretization consisting of 10 elements with the first element of length $v_1 = 1.5$. In each simulation a different value of Δ was used to refine this coarse discretization. The number of elements comprising the final discretization, the rL_2 values and the CPU requirements to solve the problem (including solution of the problem over the coarse discretization) were recorded. Values of Δ were decreased geometrically from $\Delta = 0.2$ (which resulted in a discretization of 11 elements) to $\Delta = 0.0001$ (which resulted in a discretization of 59 elements). In each case the automatic refinement procedure was applied only once.

Two series of simulations were performed to investigate the effects of geometric refinement and automatic refinement on the rL_2 error and CPU requirements for the aggregation problem. The aggregation problem was solved for a time constant $\tau = 10$ over the domain $v \epsilon (0, 170)$. Over this smaller domain there is no need to use the asymptotic approximation of the solution which was seen in figure 4.5 to significantly affect the REN values.

In the first series of simulations the PBE for aggregation was solved over a domain that was discretized and refined according to a geometric discretization. The initial discretization consisted of ten elements with a first element of length $v_1 = 1.5$ units. The number of elements was increased at each refinement stage (according to the same geometric progression) by increments of five elements until a discretization of 60 elements was obtained. The rL_2 value and CPU requirements for each discretization were recorded.

In the second series of simulations the same problem was solved but the automatic procedure was used to discretize and refine the domain. In each case a solution was obtained over a coarse discretization consisting of 10 elements with the first element of length $v_1 = 1.5$. In each simulation a different value of Δ was used to refine this coarse discretization. The number of elements comprising the final discretization, the rL_2 values and the CPU requirements to solve the problem (including solution of the problem over the coarse discretization) were recorded. Values of Δ were decreased geometrically from $\Delta = 0.2$ (which resulted in a discretization of 11 elements) to $\Delta = 0.0001$ (which resulted in a discretization of 59 elements). In each case the automatic refinement procedure was applied only once.

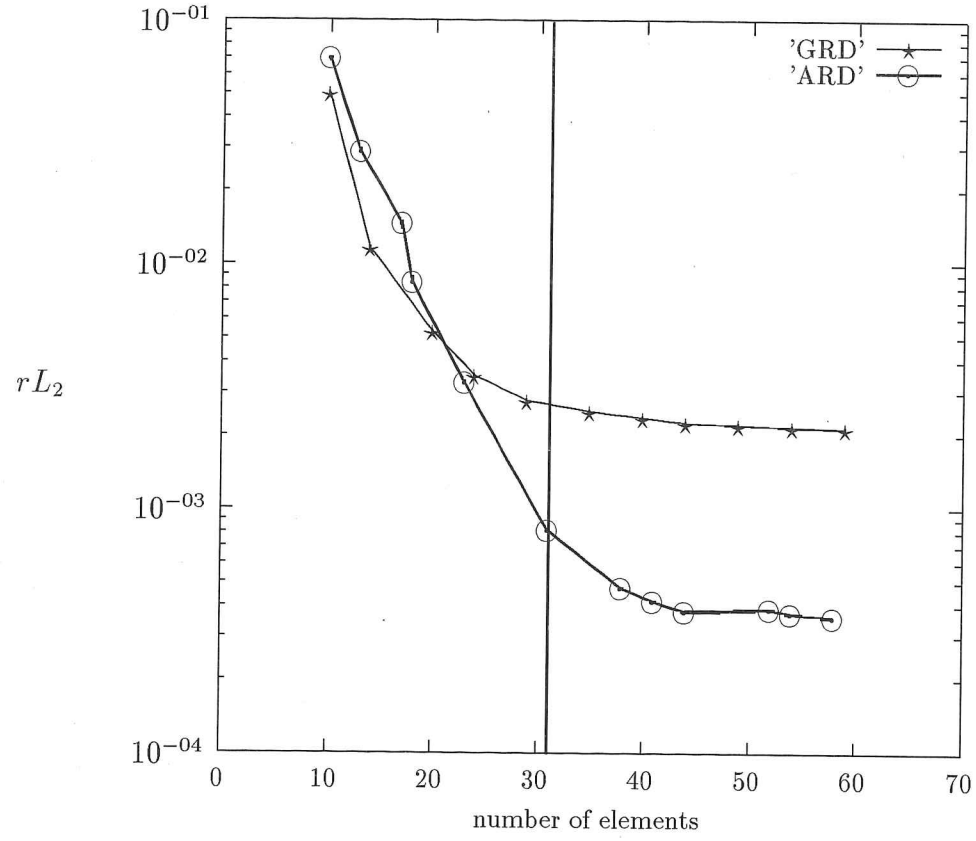

Figure 4.7: A comparison rL_2 values for numerical solutions to the aggregation problem over geometrically refined domains (GRD) and automatically refined domains (ARD).

Figure 4.7 shows rL_2 values for each simulation as a function of the number of elements comprising the discretization. The rL_2 values obtained over geometrically refined domains (GRD) are shown as stars ($\star$) while those obtained over automatically refined domains are shown as odots ($\odot$). In this figure it can be seen that for discretizations consisting of more than 20 elements the ARD achieves higher quality solutions (solutions with lower rL_2 values) than a GRD composed of the same number of elements. The vertical line corresponds to the number of elements comprising the automatic discretization when an error tolerance $\Delta = 0.05$ was used.

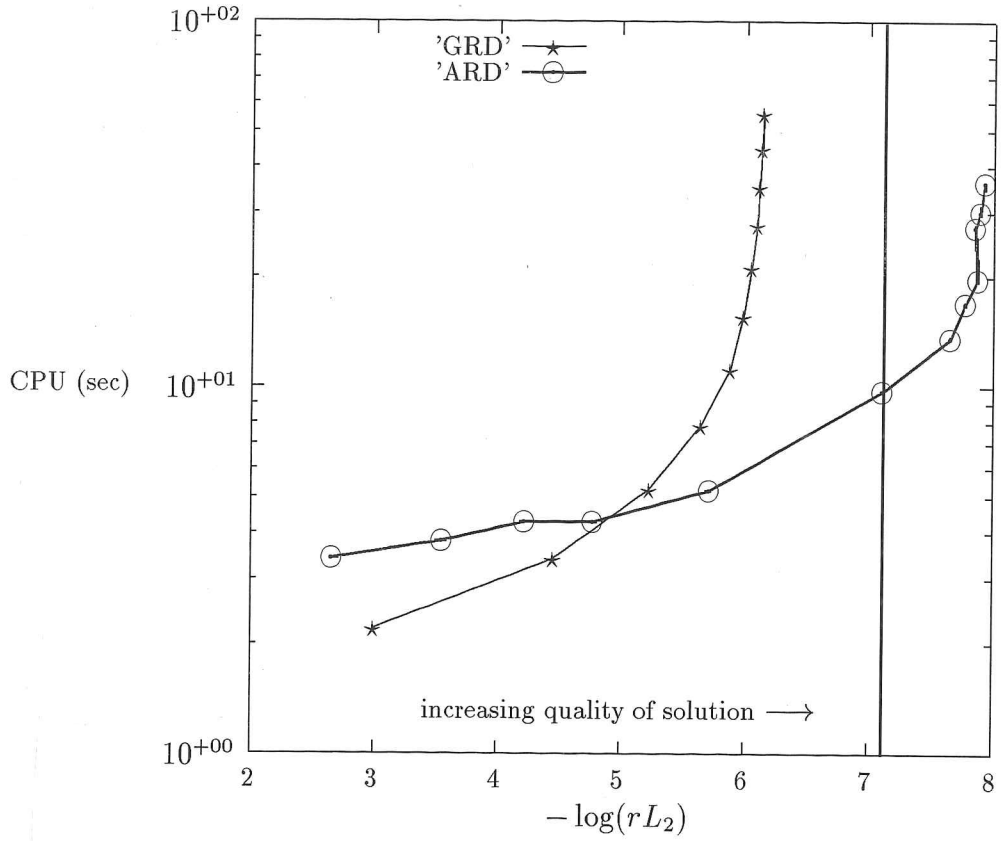

Figure 4.8: A comparison of the computation cost of a solution to the aggregation problem as a function of its quality over geometrically refined domains (GRD) and automatically refined domains (ARD).

Figure 4.8 show the cost of obtaining a numerical solution as a function its "quality". The quality of each numerical solution is quantified as :

$$\text{solution quality} := -\log(rL_2) \tag{4.14}$$

Solution quality has been defined in this manner so that better quality solutions are progressively approached from the left. Points corresponding to solutions over geometrically refined domains are once again represented by stars ($\star$) while those over automatically refined domains are represented by odots ($\odot$). In this figure it can be seen that high solution qualities $-\log(rL_2) > 5$ are achieved in significantly less time over ARDs than is achieved over GRDs. The vertical line corresponds to the

quality of solution obtained over the ARD when an error tolerance of $\Delta = 0.05$ was used.

4.7.3 A Breakage Problem

The automatic mesh refinement procedure was applied to the steady state PBE for breakage (case 2a). The parameters of this problem are summarized below :

PBE under study

$$\frac{n(v) - n_{in}(v)}{\tau} = \int_v^\infty \rho(v,w)S(w)n(w)dw - n(v)S(v)$$

$$\text{where : } \quad n_{in}(v) = \exp(-v), \quad \rho(v,w) = \frac{2}{w}, \quad \tau = 1$$

Domain considered : $v \in (0, 15.629)$

Initially this problem was solved over a geometrically discretized and refined domain :

Summary of Results over a Geometric Discretization :

Number of elements used in the discretization = 15

Length of first element = 0.1 unit

Number of iterations required for convergence = 15

Resulting rL_2 value = 0.0914

The automatic refinement procedure was then applied to the problem over the above mentioned geometric discretization. The values of Δ was altered until a final discretization of 15 elements was obtained. This was done to ascertain whether (and at what additional cost) the automatic refinement procedure could be used to improve the overall quality of a solution over a geometric discretization.

Summary of Results over an Automatically Refined Discretization :

Δ value = 0.08

Expression used to calculate new element lengths : (4.12)

Number of elements in the final discretization = 15

Number of applications of the refinement procedure = 3

Additional number of iterations required = 24

Final rL_2 value over the automatically refined domain = 0.00443

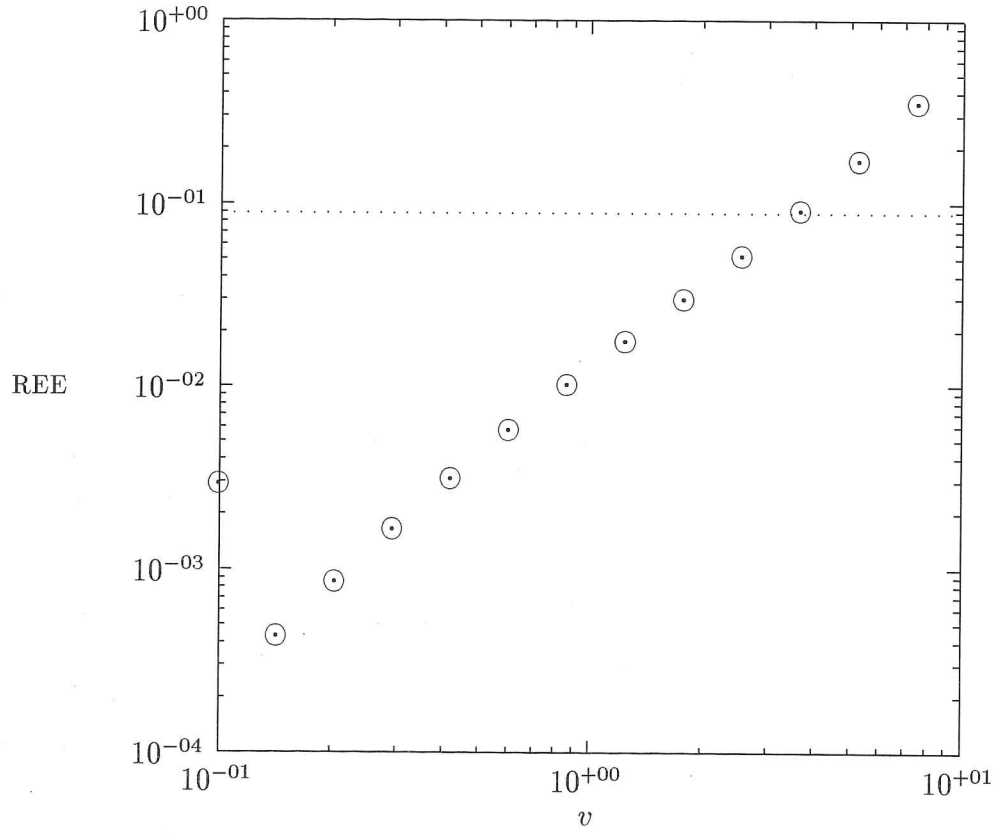

Figure 4.9: REE values for breakage problem 2a over a geometrically discretized domain

The refinement procedure progressed in a similar manner as in the aggregation problem. However, in this case three applications of the procedure were required before all of the REE values were reduced below the pre-specified tolerance $\Delta = 1.1 \times 0.08 = 0.088$. The changes in the REE values are shown in figures 4.9 to 4.12 for each

step of the refinement procedure. Although the problem was solved over the domain $v \in (0, 15.629)$ only the region $v \in (0, 9.237)$ needs to be considered to calculate the first two moments accurately. The problem was solved over the larger domain so that truncation errors in the the birth rate due to breakage could be reduced to sufficiently small values. The results of this case study will only be shown over the smaller domain $v \in (0, 9.237)$.

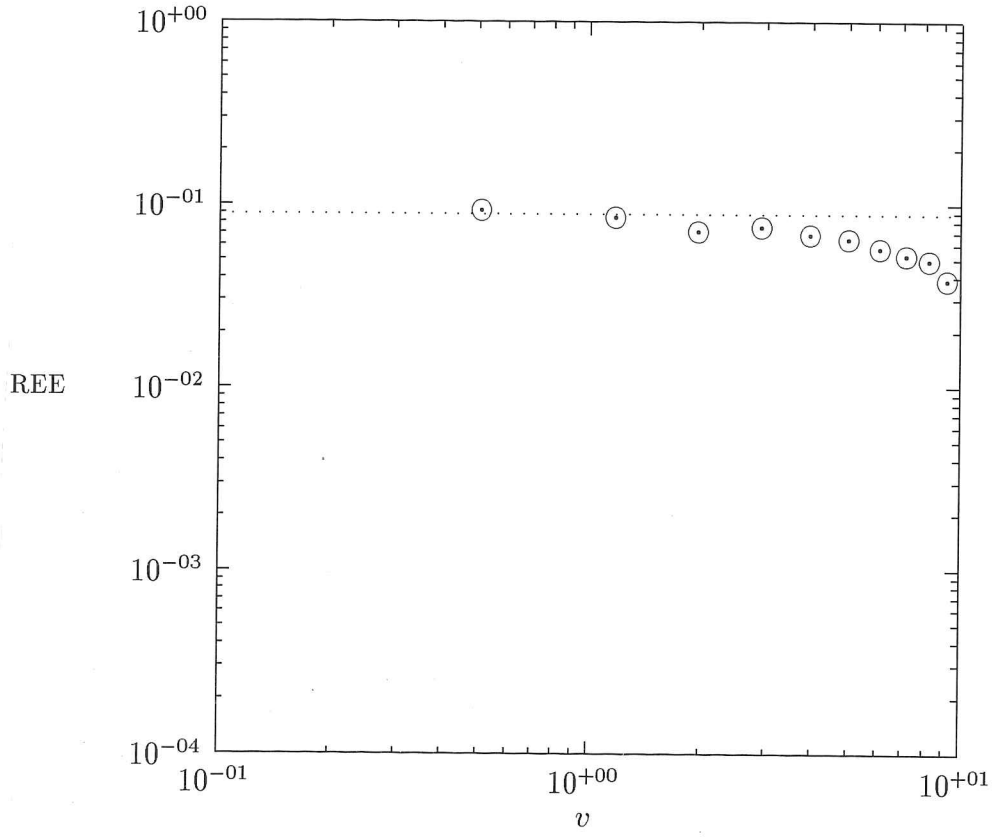

Figure 4.10: REE values for breakage problem 2a after one application of the automatic refinement procedure

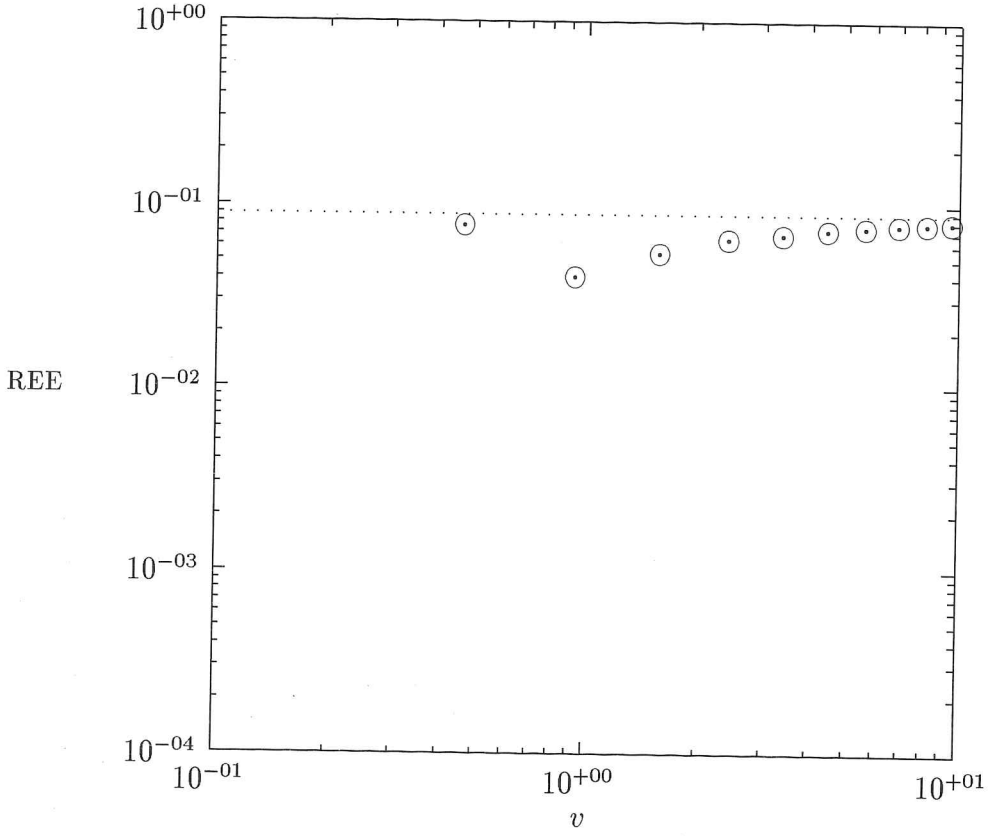

Figure 4.11: REE values for breakage problem 2a after two applications of the automatic refinement procedure

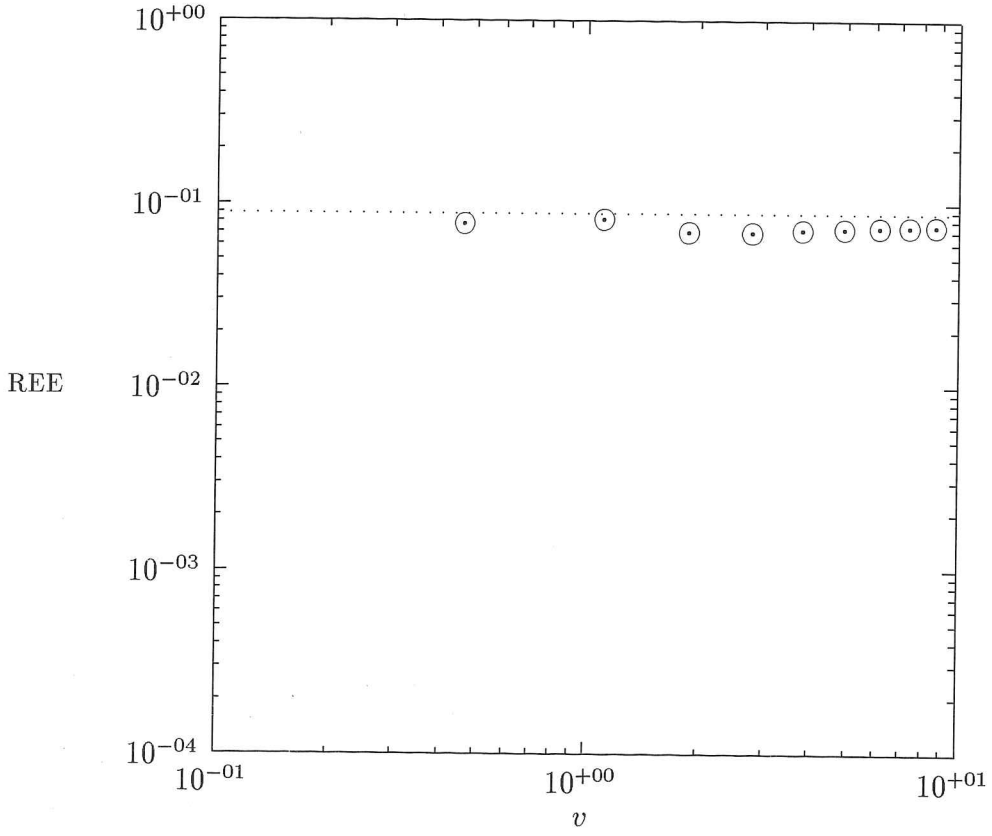

Figure 4.12: REE values for breakage problem 2a upon completion of the automatic refinement procedure

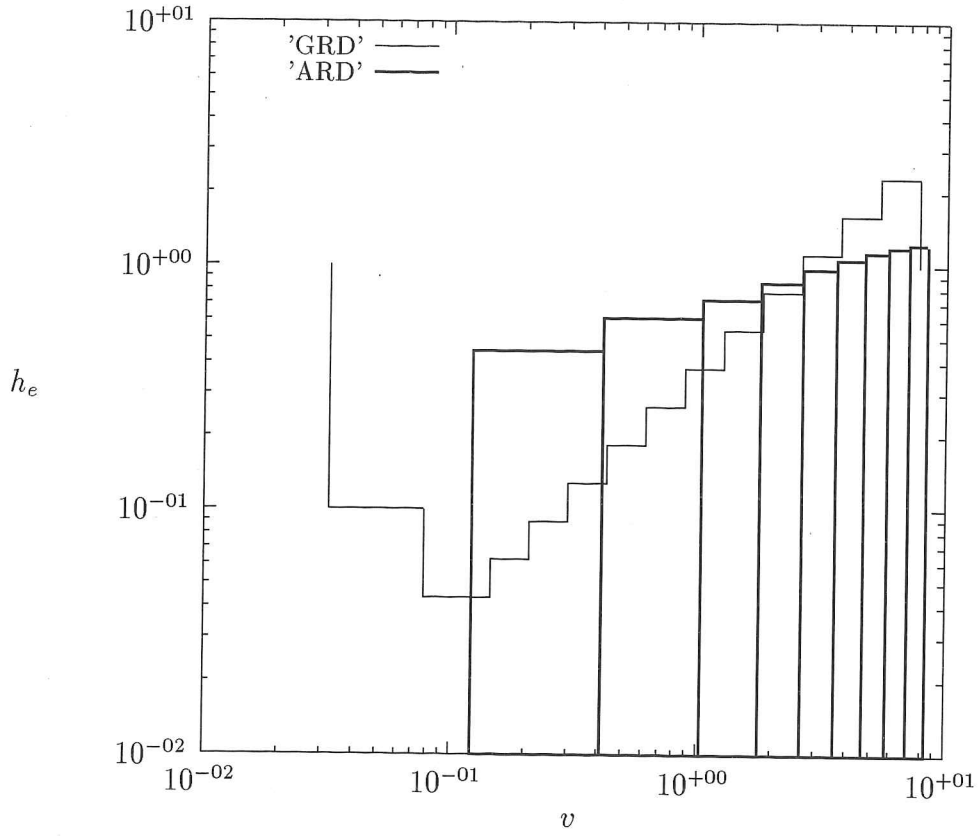

Figure 4.13: Element lengths over a geometrically refined discretization (GRD) and over an automatically refined discretization (ARD) for breakage problem 2a

Figure 4.13 shows the element lengths comprising the geometrically and automatically refined discretizations. The element lengths of the GRD are shown as a step function while those of the ARD are shown as boxes. As in the previous case, the tail region of the ARD is spanned by smaller elements than that of the GRD and *vice versa* for the rest of the domain.

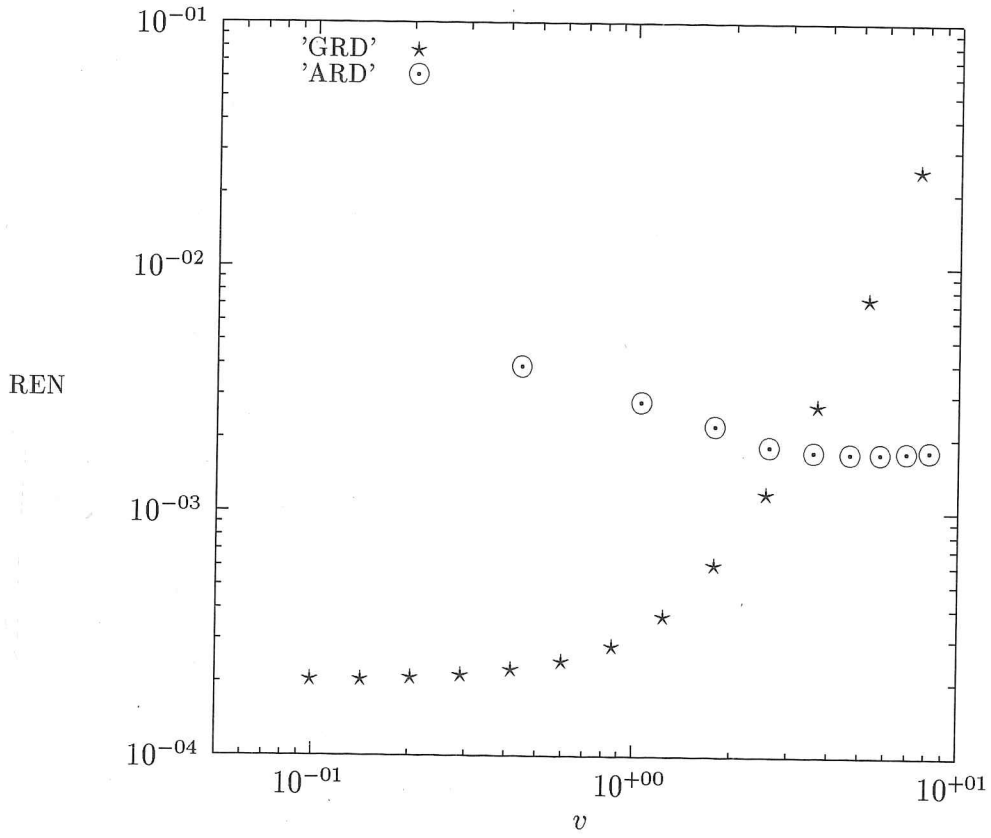

Figure 4.14: REN values over a geometrically refined discretization and over the automatically refined discretization for breakage problem 2a

Figure 4.14 shows REN values over the geometrically and automatically refined domains as stars ($\star$) and odots ($\odot$) respectively. In this case the REN values for the ARD are visibly more uniform than those obtained over a GRD.

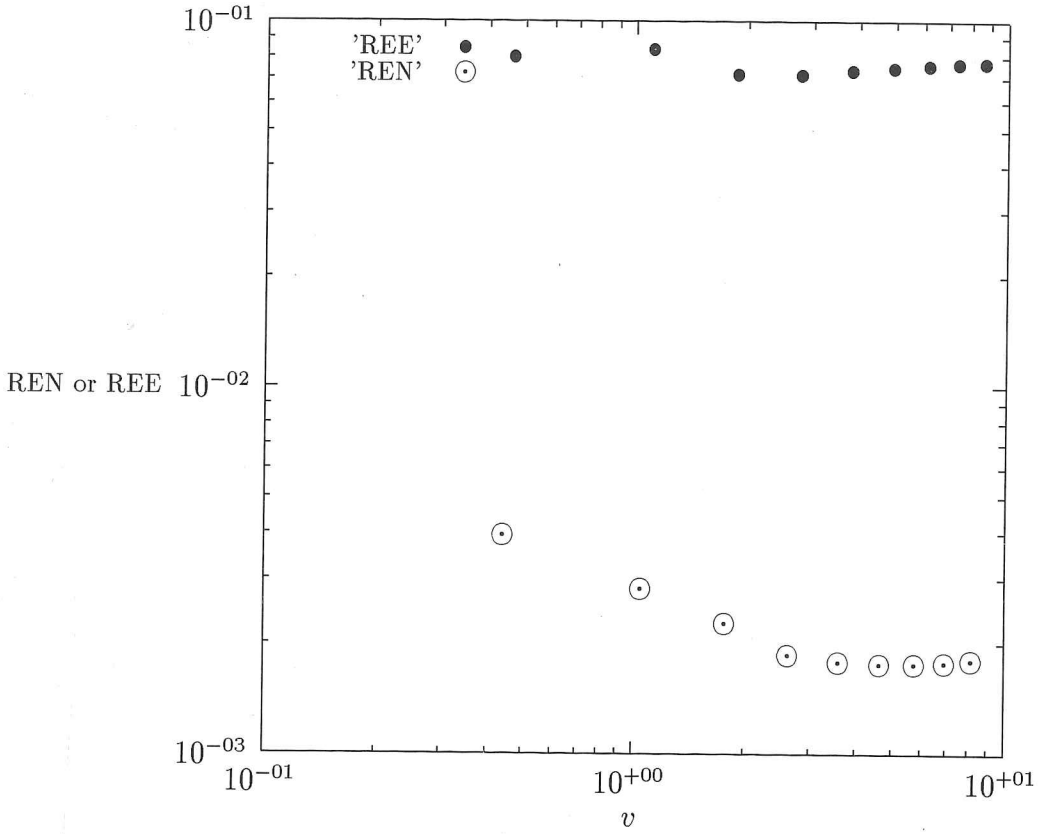

Figure 4.15: REN and REE values over the automatically refined discretization for breakage problem 2a

The REE and REN values for each element of the automatically refined domain were calculated and are plotted in figure 4.15. The REE values bound the REN values from above, hence estimate (‡) is still valid over the automatically refined domain.

As in the aggregation problem a large number of simulations were performed to compare the convergence properties of the finite element method over geometrically and automatically discretized domains. In the case of geometrically discretized domains the first element length was held constant at $h_1 = 0.1$ and the number of elements was increased according to a geometric progression from 15 to 80 elements. While in the case of automatic discretization, refinement of a coarse geometric discretization of 10 elements was performed. A different Δ value was used in each simulation. This value was adjusted so that final discretizations consisted of between 7 and 76 elements after

one application of the automatic refinement algorithm. In each case the rL_2 value and the CPU time required for convergence were recorded.

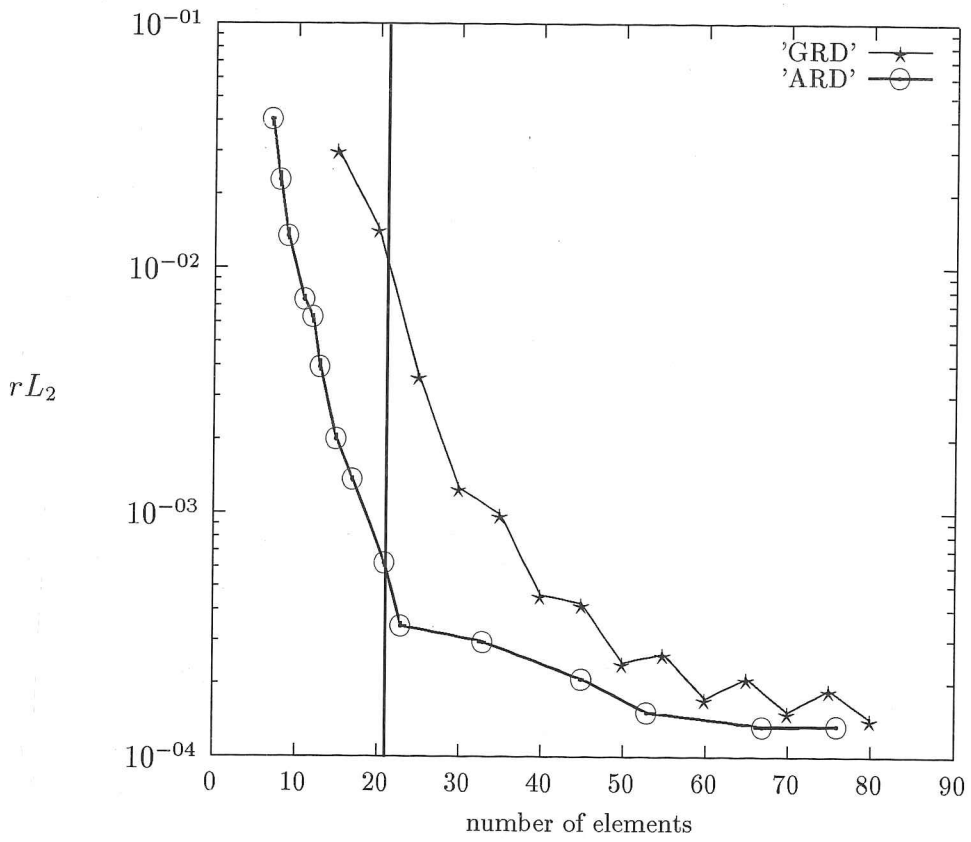

Figure 4.16: A comparison of the rL_2 values as functions of the number of elements spanning the domain for the breakage problem over geometrically refined domains (GRD) and automatically refined domains (ARD).

Figure 4.16 shows rL_2 values as a function of the number of elements comprising a domain for several simulations over geometrically and automatically refined domains. The rL_2 values obtained over geometrically refined domains (GRD) are shown as stars ($\star$) while those obtained over automatically refined domains are shown as odots ($\odot$). In every simulation, solutions obtained over ARDs were of higher quality (*ie.* had lower rL_2 values) than those obtained over GRDs consisting of the same number of elements. The vertical line corresponds to the number of elements comprising the automatically discretized domain when an error tolerance of $\Delta = 0.05$ is used.

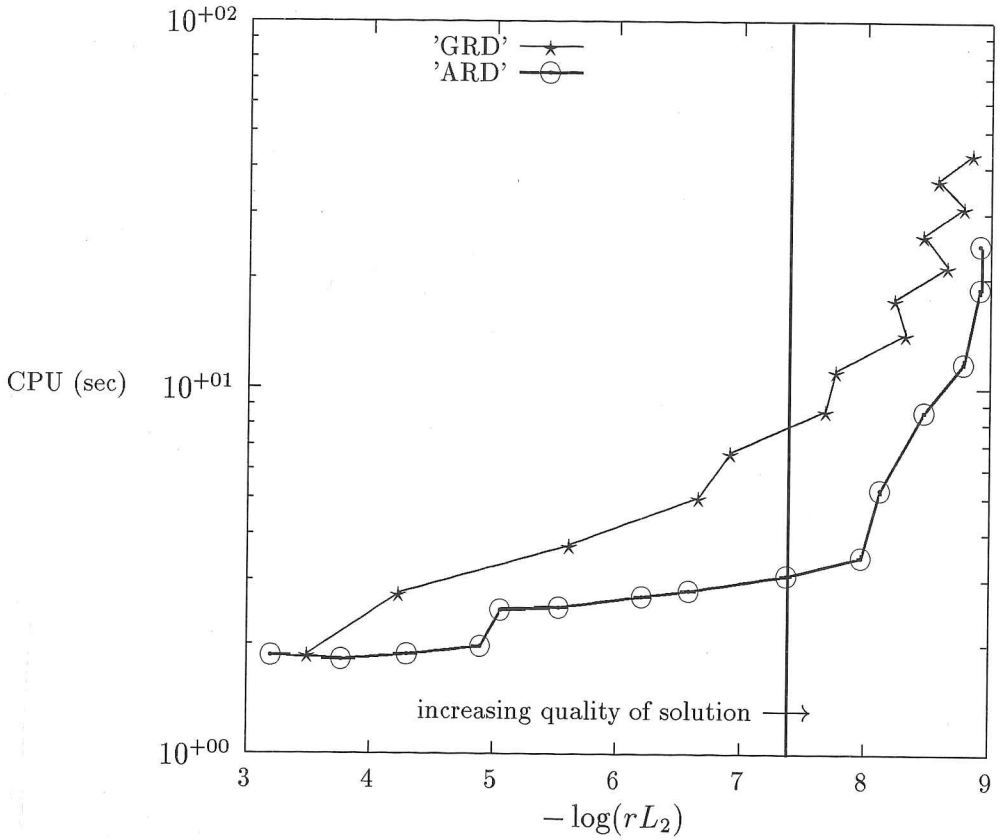

Figure 4.17: A comparison of the computational cost of a solution to the breakage problem as a function of its quality over geometrically refined domains (GRD) and automatically refined domains (ARD).

In figure 4.17 the cost of obtaining a numerical solution is plotted as a function of its quality. The quality of a numerical solution is as defined in the previous case study while the cost is the CPU requirements in seconds. Points corresponding to solutions over geometrically refined domains are once again represented by stars ($\star$) while those over automatically refined domains are represented by odots ($\odot$). From this figure it can be seen that a pre-specified solution quality would be obtained in less time over an ARD than would be required over a GRD. The vertical line corresponds to the quality of solution obtained over the automatically refined domain when an error tolerance of $\Delta = 0.05$ is used.

4.7.4 A Growth Problem

The automatic mesh refinement procedure was applied to the steady state PBE for growth (case 3a). The parameters of this problem are summarized below :

PBE under study

$$\frac{n(v) - n_{in}(v)}{\tau} + \frac{d}{dv}\Big(G(v)n(v)\Big) = 0$$

where : $n_{in}(v) = v \times \exp(-v), \quad G(v) = G_0 = 0.1, \quad n(0) = 0, \quad \tau = 1$

Domain considered : $v \in (0, 13.69)$

Initially this problem was solved over a geometrically discretized and refined domain :

Summary of Results over a Geometric Discretization :

Number of elements used in the discretization = 20

Length of first element = 0.1 unit

Number of iterations required for convergence = 1

Resulting rL_2 value = 0.1216

Problems Encountered when Attempting to Implement Automatic Refinement in conjunction with the Galerkin Finite Element Formulation :

When the automatic refinement procedure was applied to this problem significant numerical errors were encountered. In this case the automatic refinement algorithm attempted to create very small elements ($h_e < 1 \times 10^{-4}$ *units*) in the lower volume ranges. Hence, the entries in the upper left corner of the A^g_{ij} matrix of equation (2.42) became sufficiently small to cause matrix ill-conditioning. The resulting poor solution over the automatically refined domain is shown in figure 4.18.

This problem was overcome by using a Petrov-Galerkin formulation. According to this method the weight functions are not set to the interpolation polynomials :

$$\phi^e_i(v) \neq \psi^e_i(v) \tag{4.15}$$

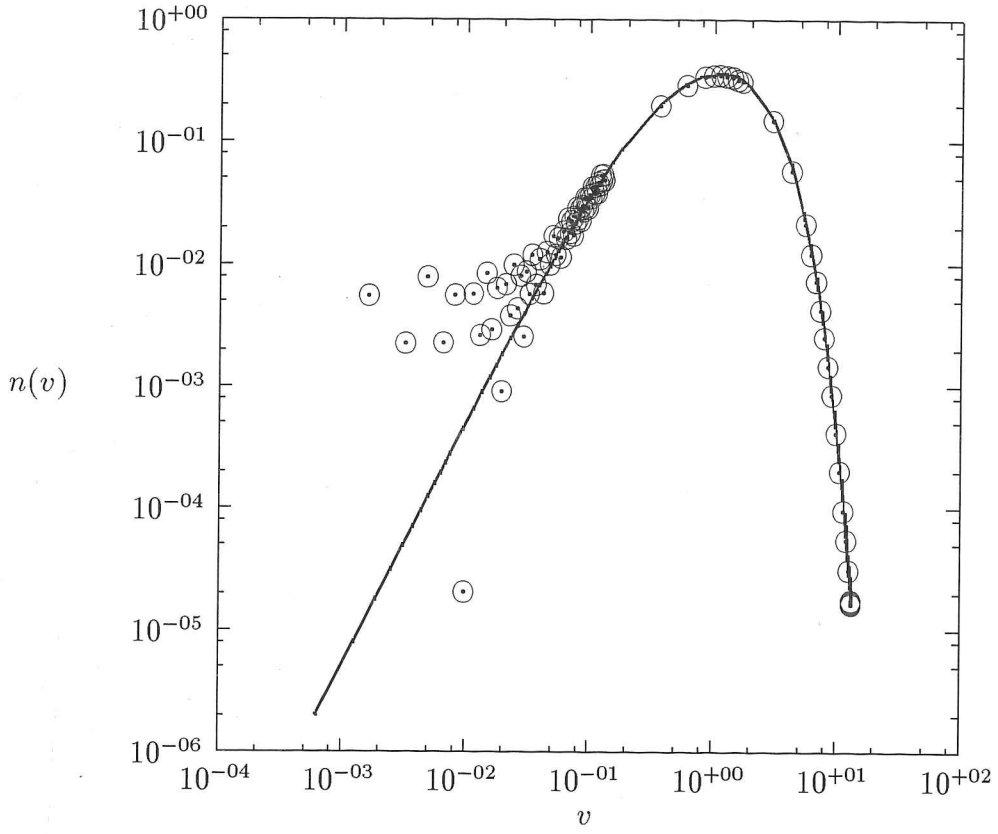

Figure 4.18: Poor solution to the growth problem 3a over an automatically refined domain. The refinement procedure created very small elements which resulted in matrix ill- conditioning. A special Petrov-Galerkin method is needed to overcome this problem.

A different set of functions is selected so that the magnitude of the contributions from small elements in the lower volume range is increased. One suitable set of functions is :

$$\phi_i^e(v) = \frac{\psi_i^e(v)}{v^{-\log(h_e)}} \tag{4.16}$$

This set of weight functions was used in all elements where $h_e < 1$ *unit* and elements were not permitted to have lengths less that 1×10^{-4} units. This formulation was found to be considerably less prone to matrix ill-conditioning and the automatic refinement algorithm could then be applied to the problem.

Summary of Results over an Automatic Discretization :

The automatic refinement procedure was then applied to the Petrov-Galerkin formulation of the problem. The above mentioned geometric discretization was used to begin the refinement procedure.

The Δ value was altered until a final discretization of 20 elements was obtained. This was done to ascertain whether (and at what additional cost) the automatic refinement procedure could be used to improve the overall quality of a solution over a geometric discretization.

Δ value $= 0.15$
Expression used to calculate new element lengths : (4.5)
Number of elements in the final discretization $= 20$
Number of applications of the refinement procedure $= 3$
Additional number of iterations required $= 3$
Final rL_2 value over the automatically refined domain $= 0.026$

Equilibration of the REE values proved more difficult than in previous cases hence the refinement procedure was terminated after 3 applications. The REE values are shown for each application of the automatic refinement procedure in figures 4.19 to 4.22.

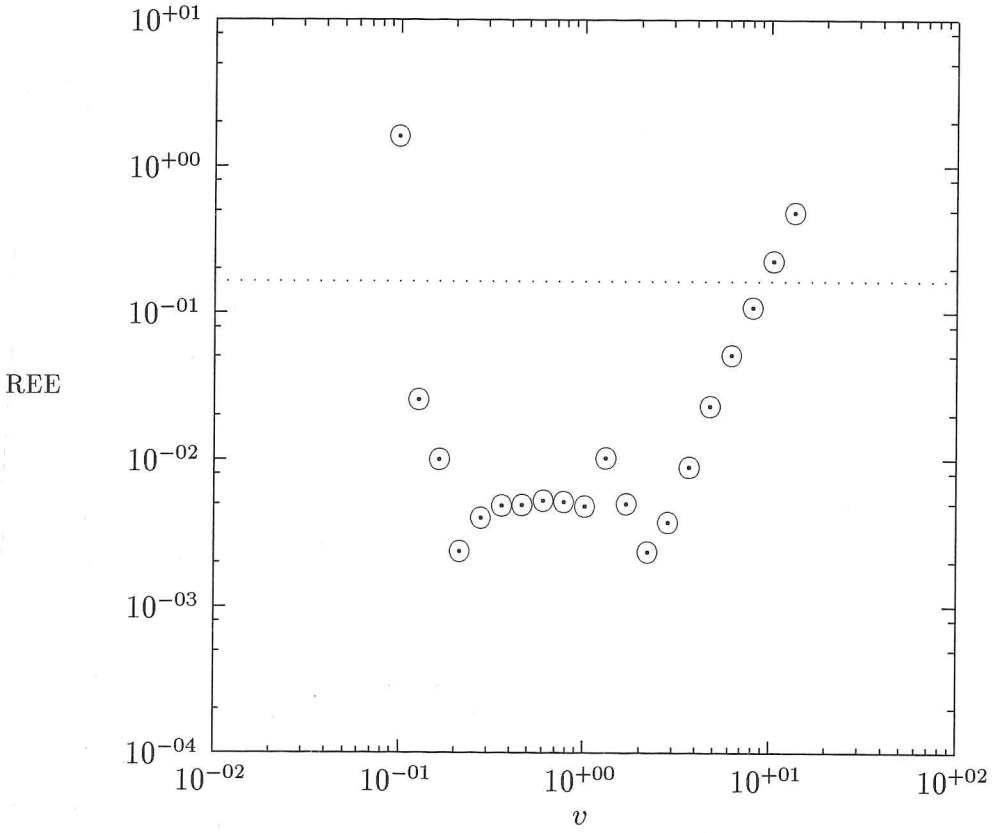

Figure 4.19: REE values for growth problem 3a over a geometrically discretized domain

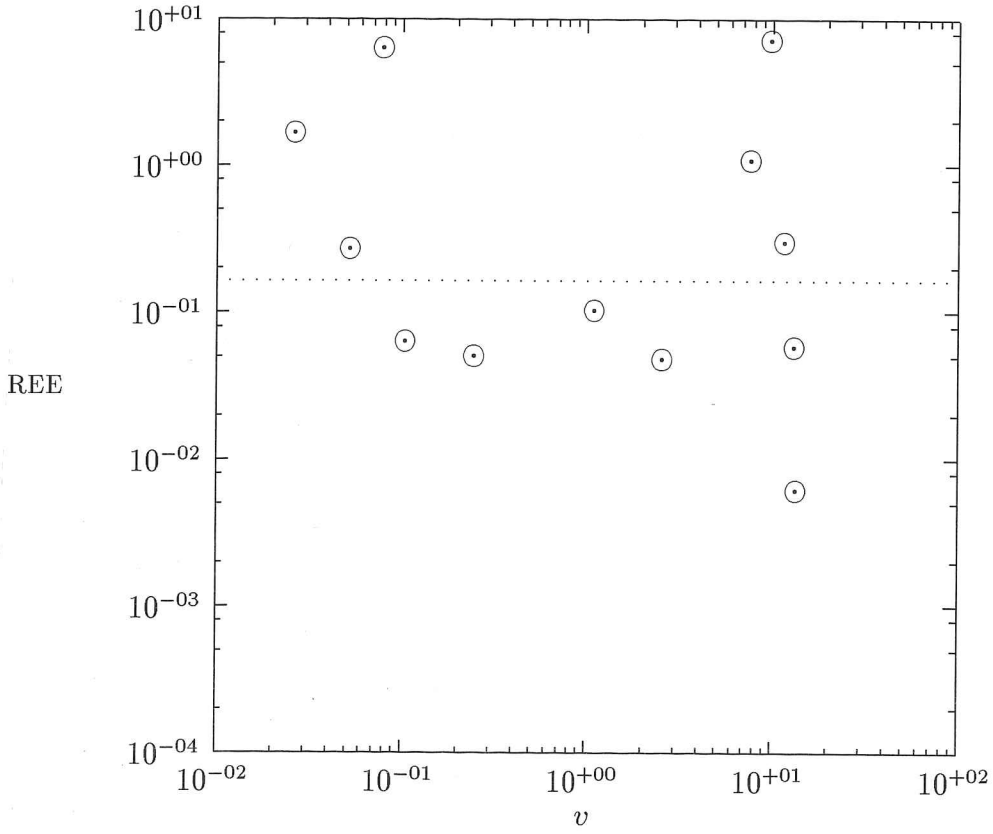

Figure 4.20: REE values for growth problem 3a after one application of the automatic refinement procedure

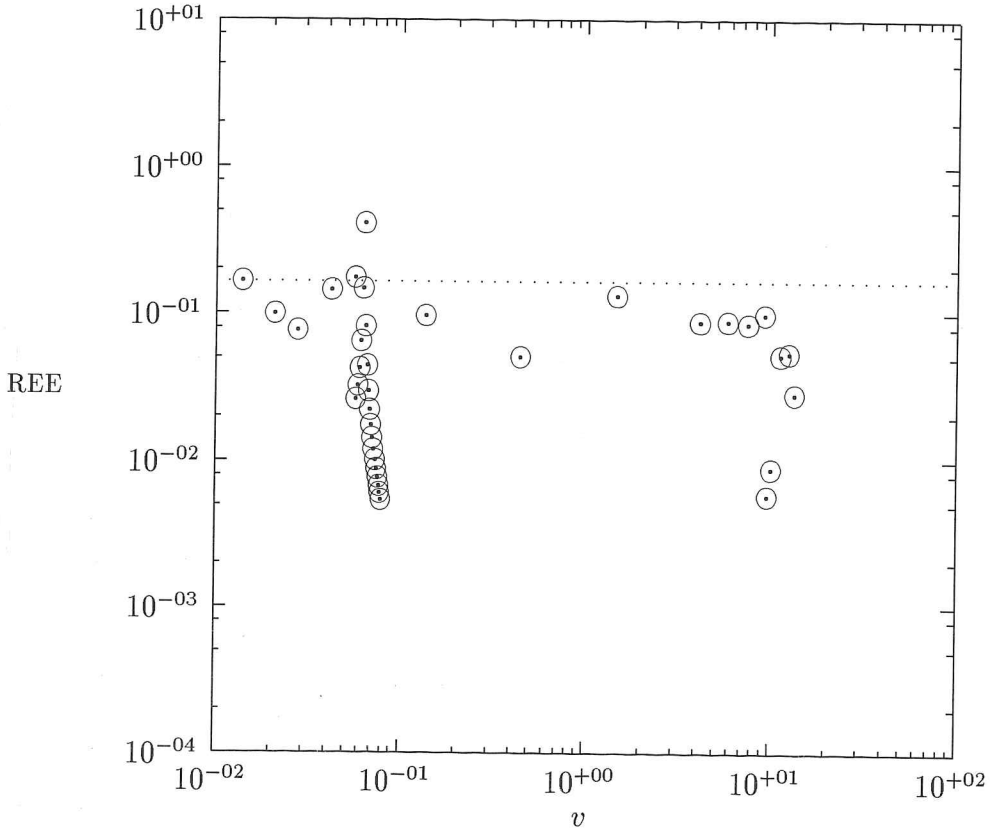

Figure 4.21: REE values for growth problem 3a after two applications of the automatic refinement procedure

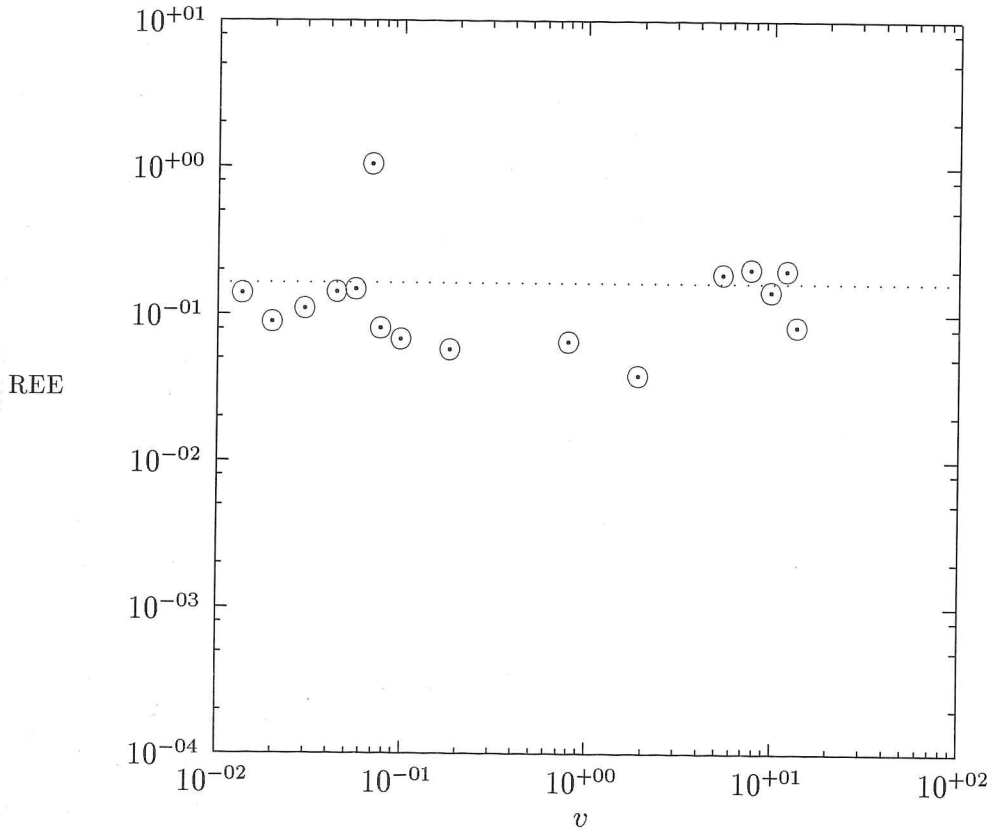

Figure 4.22: REE values for growth problem 3a after three applications of the automatic refinement procedure

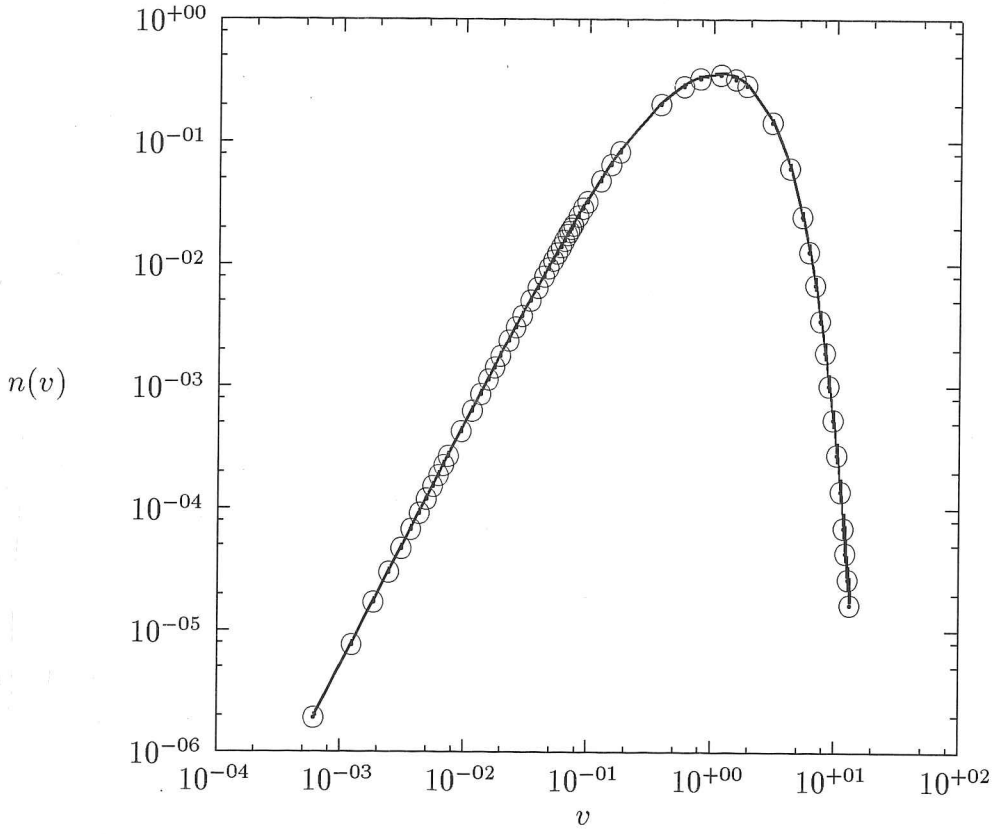

Figure 4.23: Much improved finite element predictions of the number density distribution for growth problem 3a after the automatic refinement procedure has been applied in conjunction with a Petrov Galerkin formulation.

In figure 4.23 it can be seen that the Petrov Galerkin formulation is capable of much better predictions of the density distribution for the growth problem despite the presence of very small elements that the automatic refinement algorithm has used to span some regions of the domain.

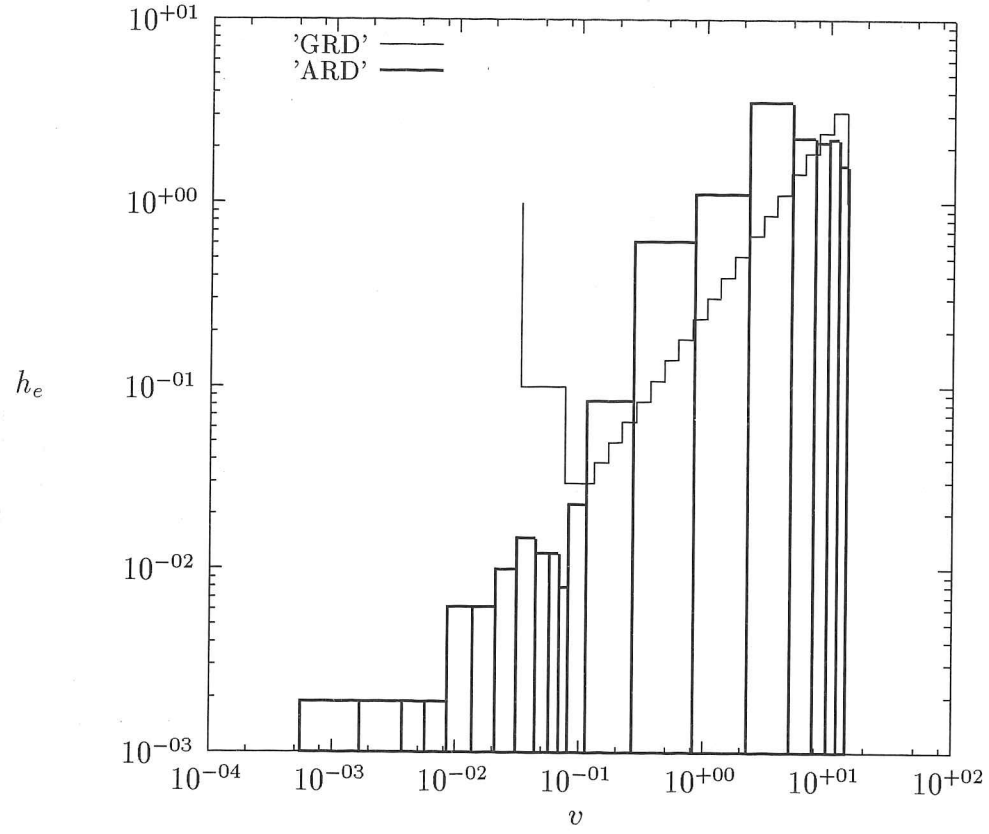

Figure 4.24: Element lengths over a geometrically refined discretization (GRD) and over an automatically refined discretization (ARD) for growth problem 3a

Figure 4.24 shows the element lengths comprising the geometrically and automatically refined discretizations. The element lengths of the GRD are shown as a step function while those of the ARD are shown as boxes. When the ARD is applied the lower size ranges are spanned by much smaller elements while the largest elements are located in the vicinity of the mode of the solution.

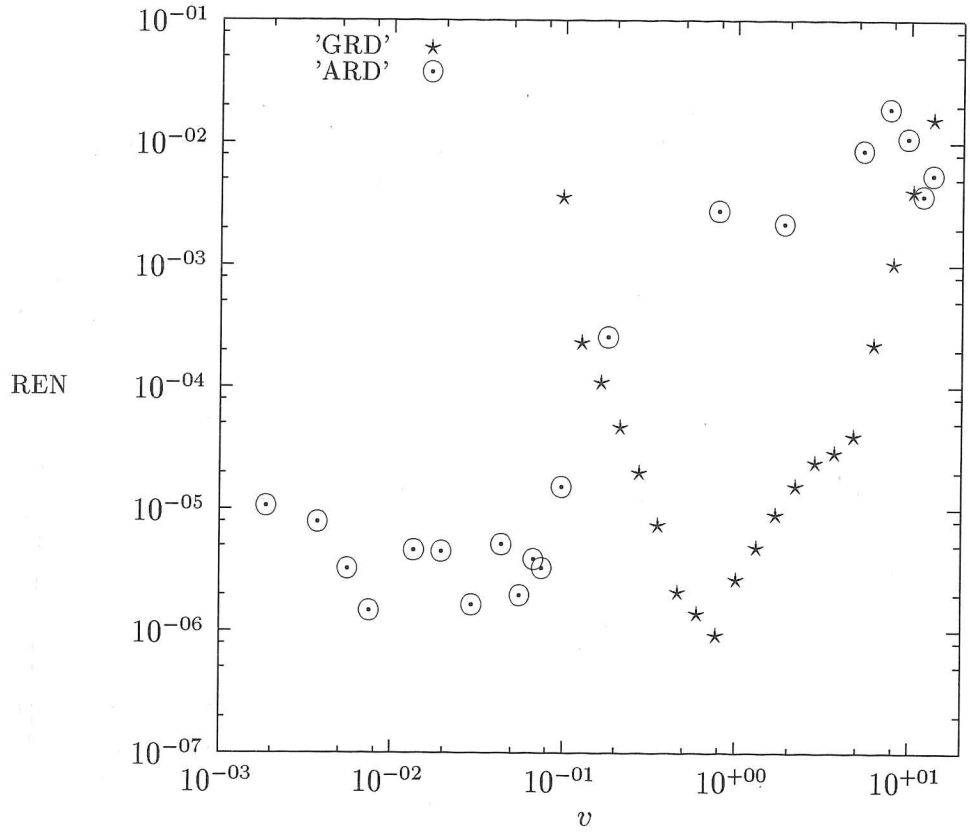

Figure 4.25: REN values over a geometrically refined discretization and over the automatically refined discretization for growth problem 3a

Figure 4.25 shows REN values over the geometrically and automatically refined domains as stars ($\star$) and odots ($\odot$) respectively. The objective of equilibrating the REN values has not been achieved by the automatic refinement algorithm. The REN values of the ARD appear to deviate as much as those of the GRD. The effect of the automatic refinement algorithm on the quality of the numerical solution will be quantified by a series of simulations performed at the end of this subsection.

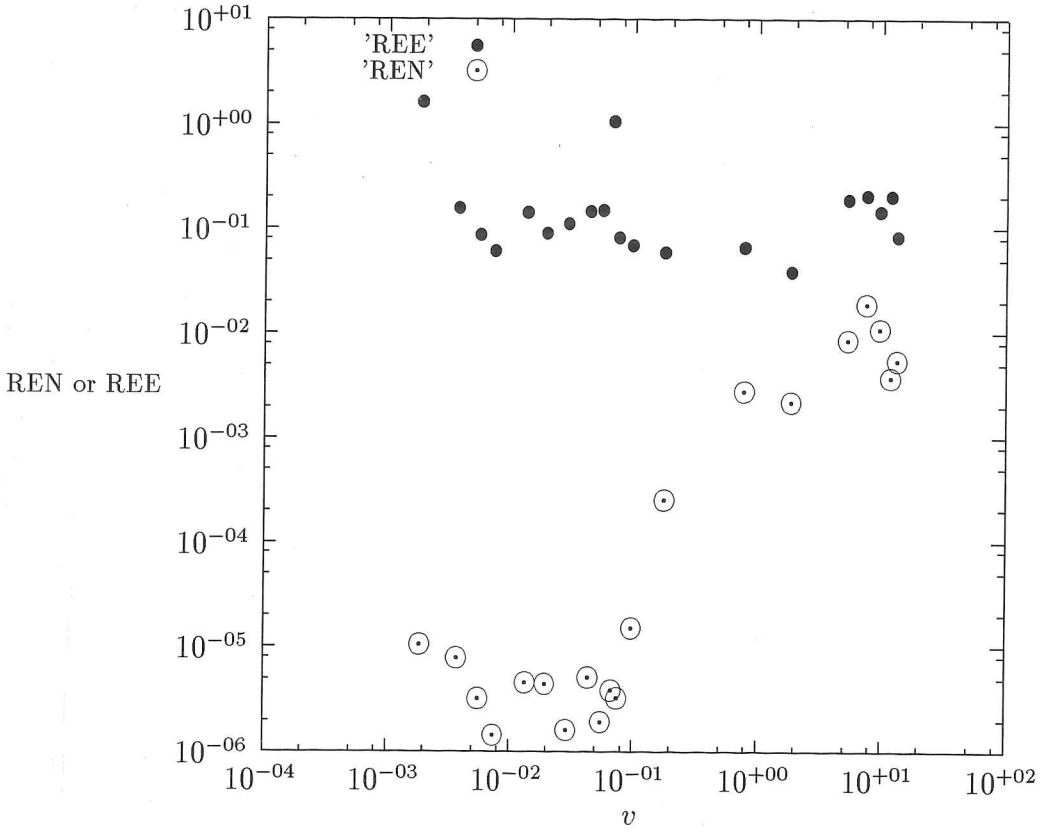

Figure 4.26: REN and REE values over the automatically refined discretization for growth problem 3a

The REE and REN values for each element of the automatically refined domain were calculated as a test of whether the automatic refinement procedure degraded the ability of estimate (†) to predict upper bounds of the errors in numerically obtained solutions. In figure 4.26 it can be seen that the REE values still bounded the REN values from above after the automatic refinement procedure was performed.

As in the previous two cases a large number of simulations were performed to compare the convergence properties of the finite element method over geometrically and automatically discretized domains. In the case of geometrically discretized domains the first element length was held constant at $h_1 = 0.1$ and the number of elements was

increased according to a geometric progression from 10 to 104 elements. While in the case of automatic discretization, refinement of a coarse geometric discretization of 10 elements was performed. A different Δ value was used in each simulation. This value was adjusted so that final discretizations consisted of between 10 and 104 elements after one application of the automatic refinement algorithm. In each case the rL_2 value and the CPU time required for convergence were recorded. Figure 4.27 shows

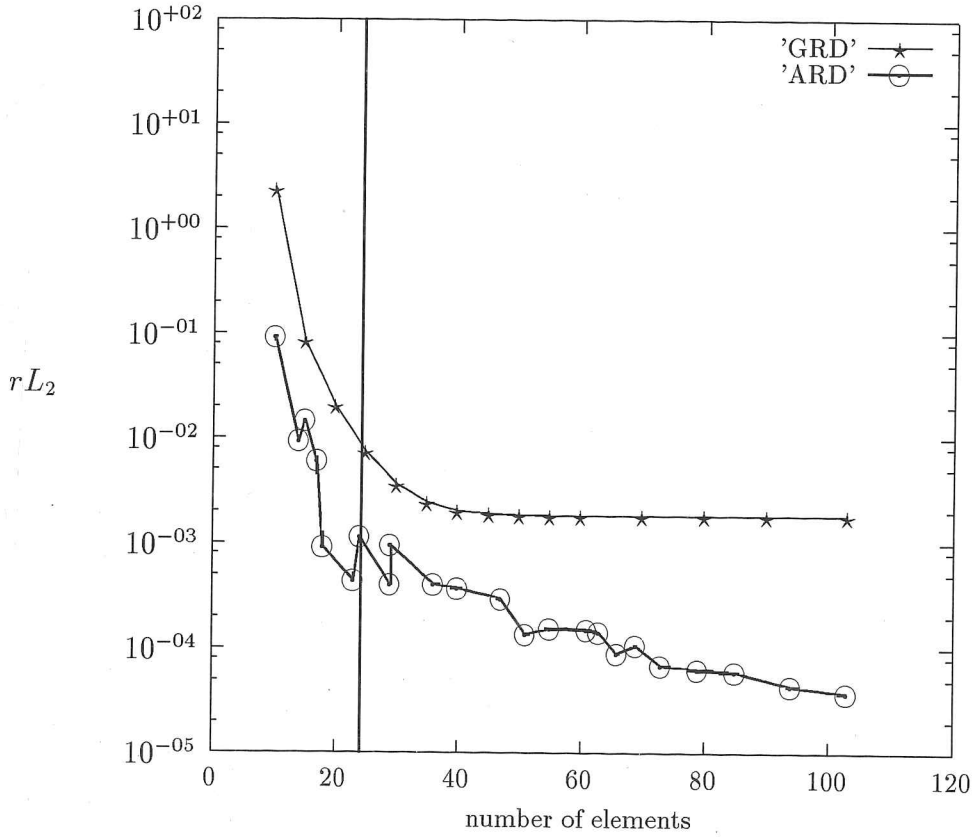

Figure 4.27: A comparison of the rL_2 values as functions of the number of elements spanning the domain for the growth problem over geometrically refined domains (GRD) and automatically refined domains (ARD).

rL_2 values as a function of the number of elements comprising a domain for several simulations over geometrically and automatically refined domains. The rL_2 values over geometrically refined domains (GRD) are shown as stars ($\star$) while those obtained over automatically refined domains are shown as odots ($\odot$). In this figure it can be

seen that higher quality solutions were obtained over the ARDs than over GRDs using the same number of elements to span the domain. Hence, even though the goal of equilibrating the REN values was not achieved to the extent originally intended, the automatic refinement algorithm still discretized the domain more efficiently than could be achieved according to a geometrical discretization.

The vertical line corresponds to the number of elements comprising the automatically refined domain when an error tolerance of $\Delta = 0.05$ is used. The significance of this line will be discussed in subsection 4.8.7

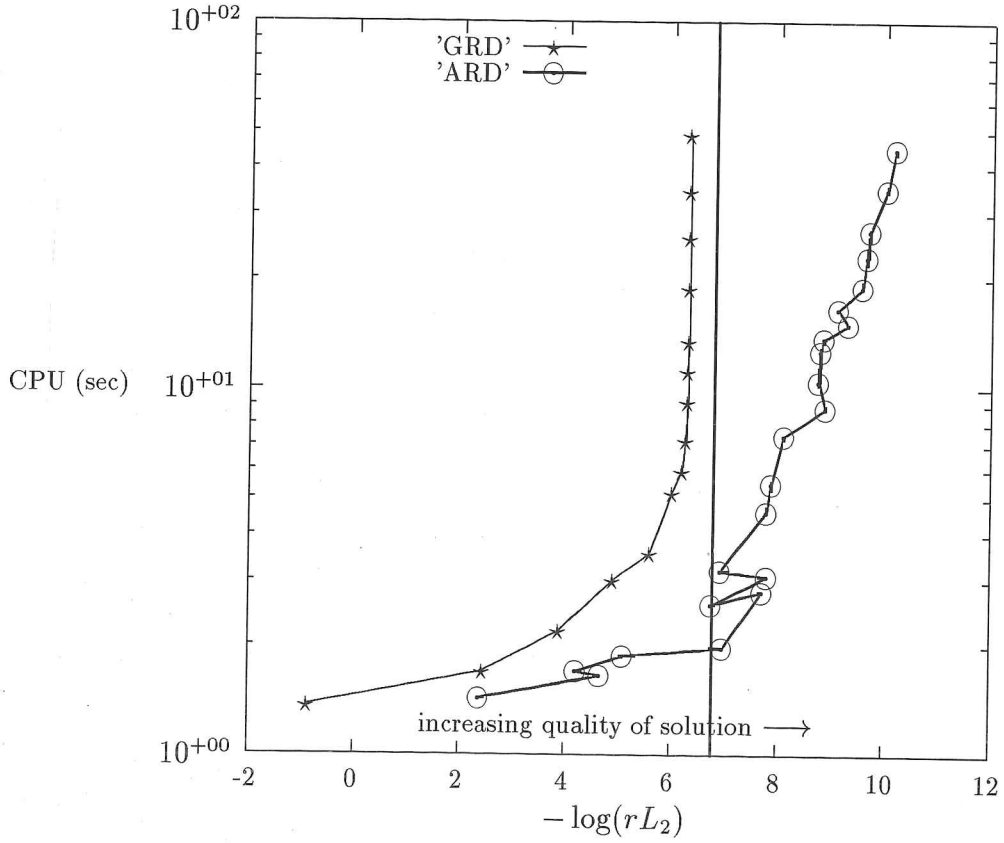

Figure 4.28: A comparison of the computation cost of a solution to the growth problem as a function of its quality over geometrically refined domains (GRD) and automatically refined domains (ARD).

In figure 4.28 the cost of obtaining a numerical solution is plotted as a function of its quality. The quality of a numerical solution is defined as in the previous case study

while the cost is the CPU requirements in seconds. Points corresponding to solutions over geometrically refined domains are once again represented by stars ($\star$) while those over automatically refined domains are represented by odots ($\odot$). In this figure it can be seen that higher quality solutions were achieved at lower cost over ARDs than over GRDs. The vertical line corresponds to the quality of solution obtained when an error tolerance of $\Delta = 0.05$ is used.

4.7.5 A Problem of Combined Aggregation, Growth and Nucleation

The automatic mesh refinement procedure was applied to the steady state PBE for aggregation, growth and nucleation (case 4a). The parameters of this problem are summarized below :

PBE under study

$$\frac{n(v) - n_{in}(v)}{\tau} + \frac{d}{dv}\Big(G(v)n(v)\Big) = \int_0^{v/2} \beta(v-w,w)n(v-w)n(w)dw - \int_0^{\infty} \beta(v,w)n(w)dw$$

where : $n_{in}(v) = 0, \quad G(v) = G_0 = 1, \quad \beta(v,w) = \beta_0 = 1, \quad \tau = 1, \quad n(0) = 1$

Domain considered : $v \in (0, 26.36)$

Initially this problem was solved over a geometrically discretized and refined domain :

Summary of Results over a Geometric Discretization :

Number of elements used in the discretization = 20

Length of first element = 0.5 unit

Number of iterations required for convergence = 24

Resulting rL_2 value = 0.011

The automatic refinement procedure was then applied to the problem over the above mentioned geometric discretization. The Δ value was altered until a final discretization of 20 elements was obtained. This was done to ascertain whether (and at what additional cost) the automatic refinement procedure could be used to improve the overall quality of a solution over a geometric discretization.

Summary of Results over an Automatically Refined Discretization :

Δ value = 0.078

Expression used to calculate new element lengths : (4.5)

Number of elements in the final discretization = 20

Number of applications of the refinement procedure = 3

Additional number of iterations required = 39

Final rL_2 value over the automatically refined domain = 0.0029

The refinement procedure progressed in a similar manner as in the aggregation problem. However, in this case three applications of the procedure were required before all of the REE values were reduced below the pre-specified tolerance $\Delta = 1.1 \times 0.078 = 0.0858$. The REE values are shown for each step of the refinement procedure in figures 4.29 to 4.32.

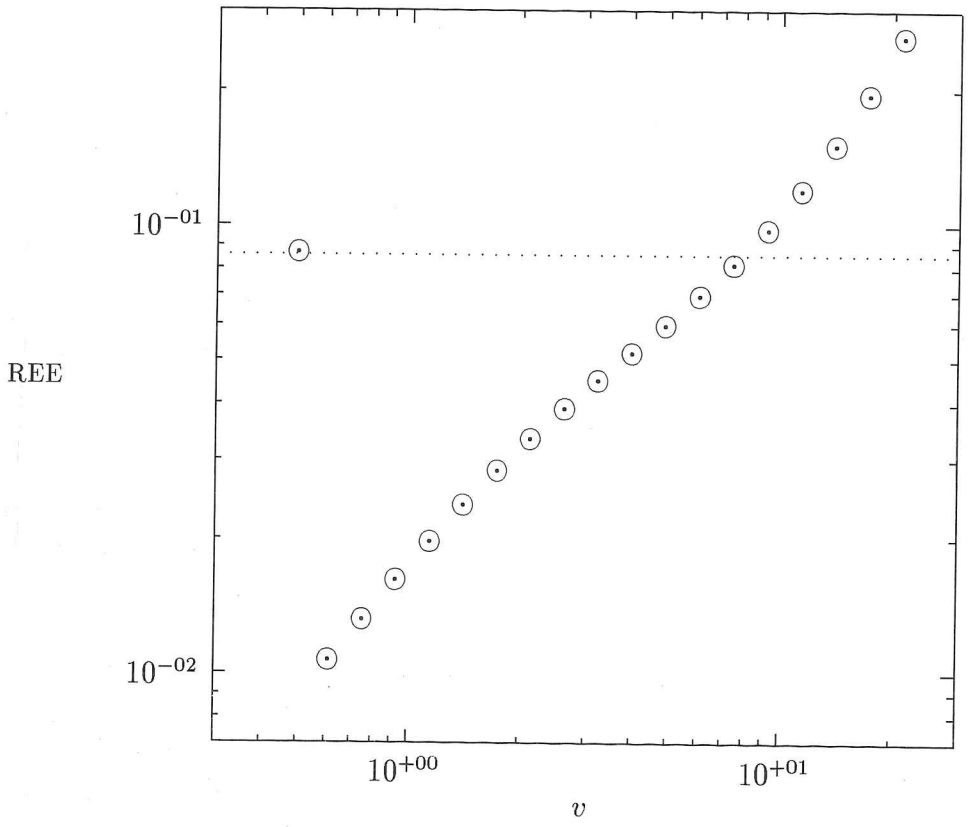

Figure 4.29: REE values for the combined aggregation, growth and nucleation problem 4a over a geometrically discretized domain

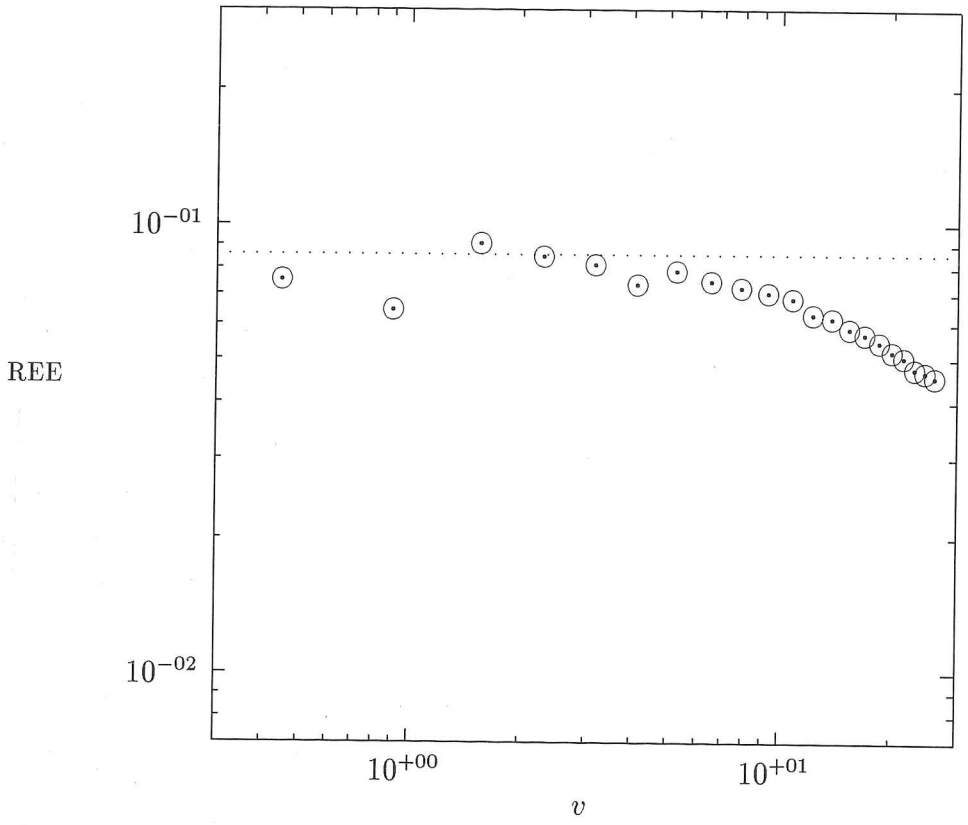

Figure 4.30: REE values for the combined aggregation, growth and nucleation problem 4a after one application of the automatic refinement procedure

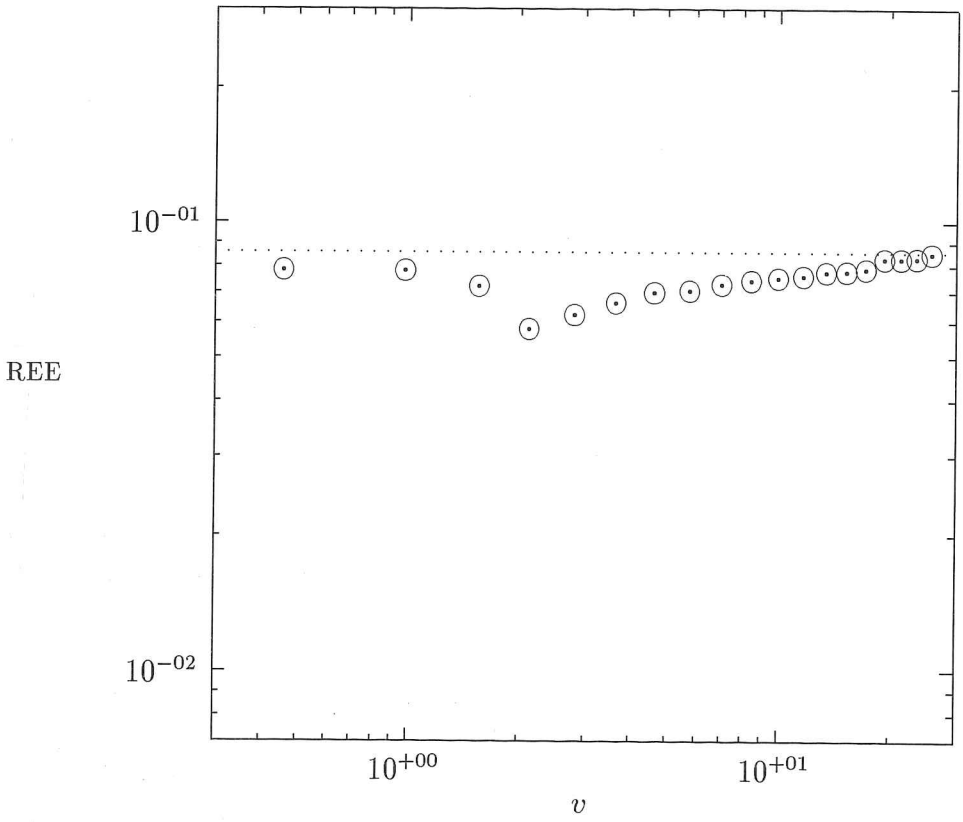

Figure 4.31: REE values for the combined aggregation, growth and nucleation problem 4a after two applications of the automatic refinement procedure

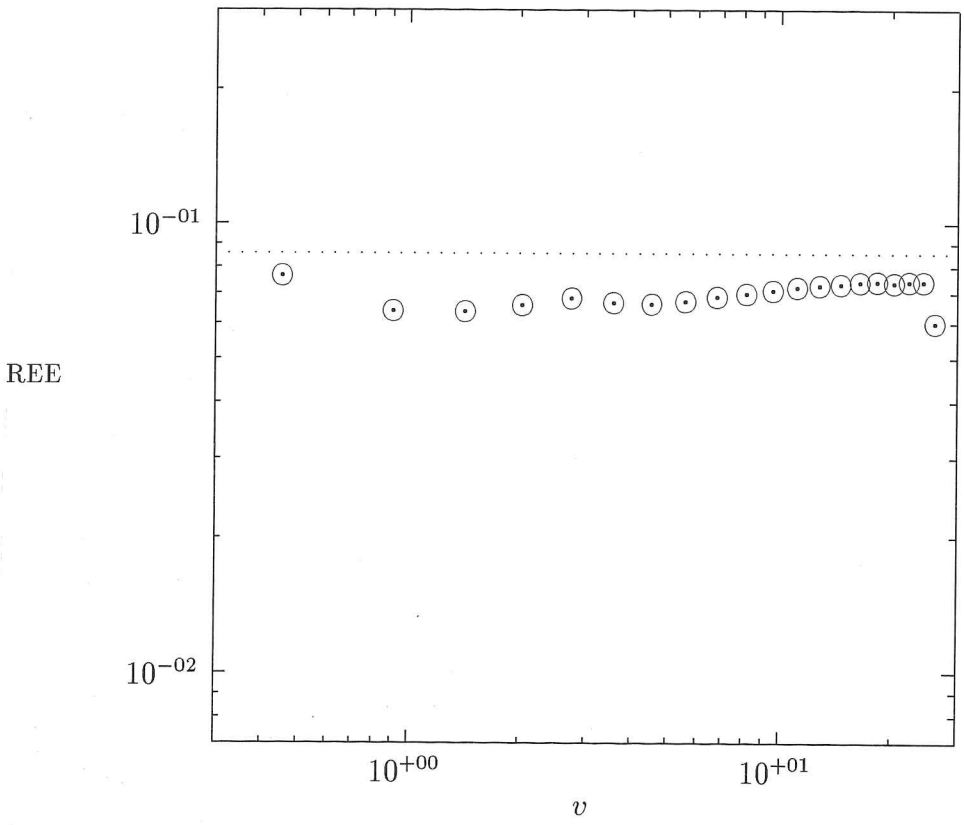

Figure 4.32: REE values for the combined aggregation, growth and nucleation problem 4a upon completion of the automatic refinement procedure

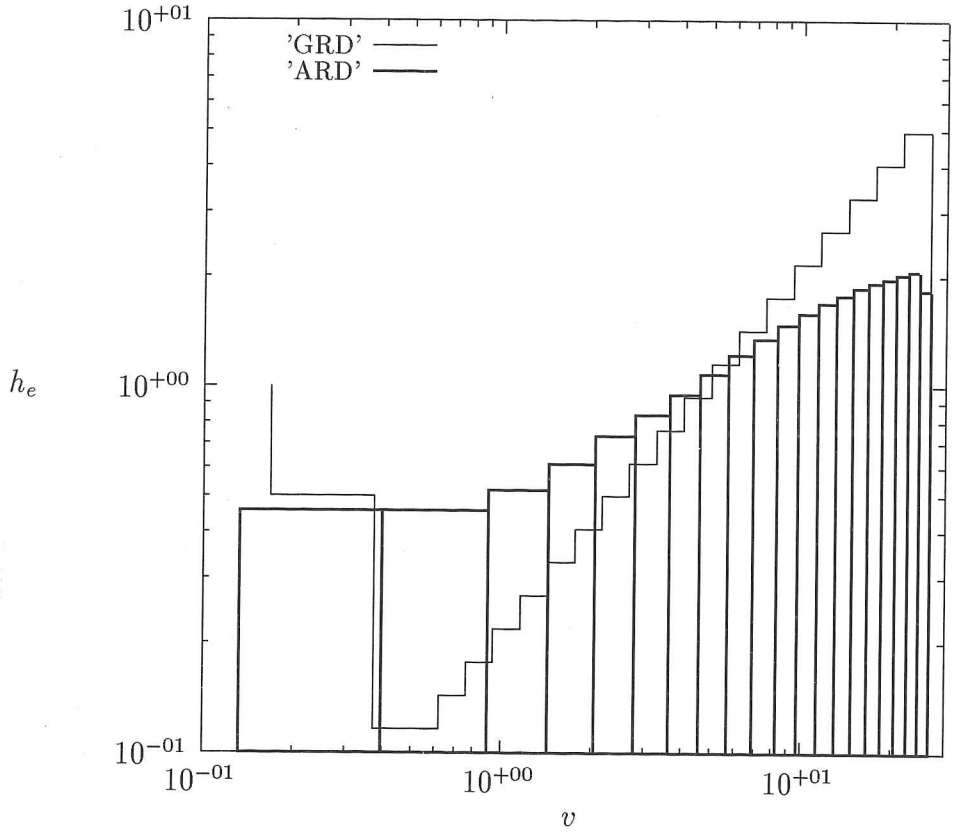

Figure 4.33: Element lengths over a geometrically refined discretization (GRD) and over an automatically refined discretization (ARD) for the combined aggregation, growth and nucleation problem 4a

Figure 4.33 shows the element lengths comprising the geometrically and automatically refined discretizations. The element lengths of the GRD are shown as a step function while those of the ARD are shown as boxes. As in the aggregation and breakage problems, the tail region of the ARD is spanned by smaller elements than those used to span the GRD and the opposite trend may be observed over the rest of the domain with the exception of the first element.

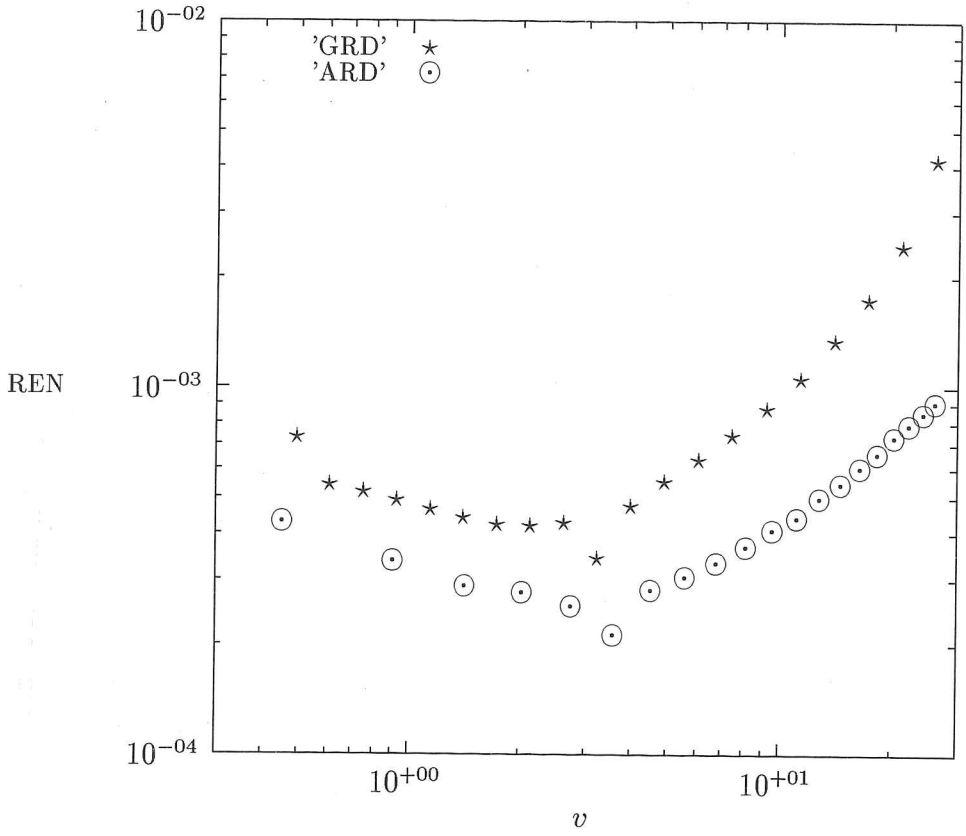

Figure 4.34: REN values over a geometrically refined discretization and over the automatically refined discretization for the combined aggregation, growth and nucleation problem 4a

Figure 4.34 shows REN values over the geometrically and automatically refined domains as stars ($\star$) and odots ($\odot$) respectively. The REN values of the ARD appear to be slightly more uniform than those of the GRD. The effect of automatic refinement on the quality of numerical solutions will be quantified by the series of solutions performed at the end of this section.

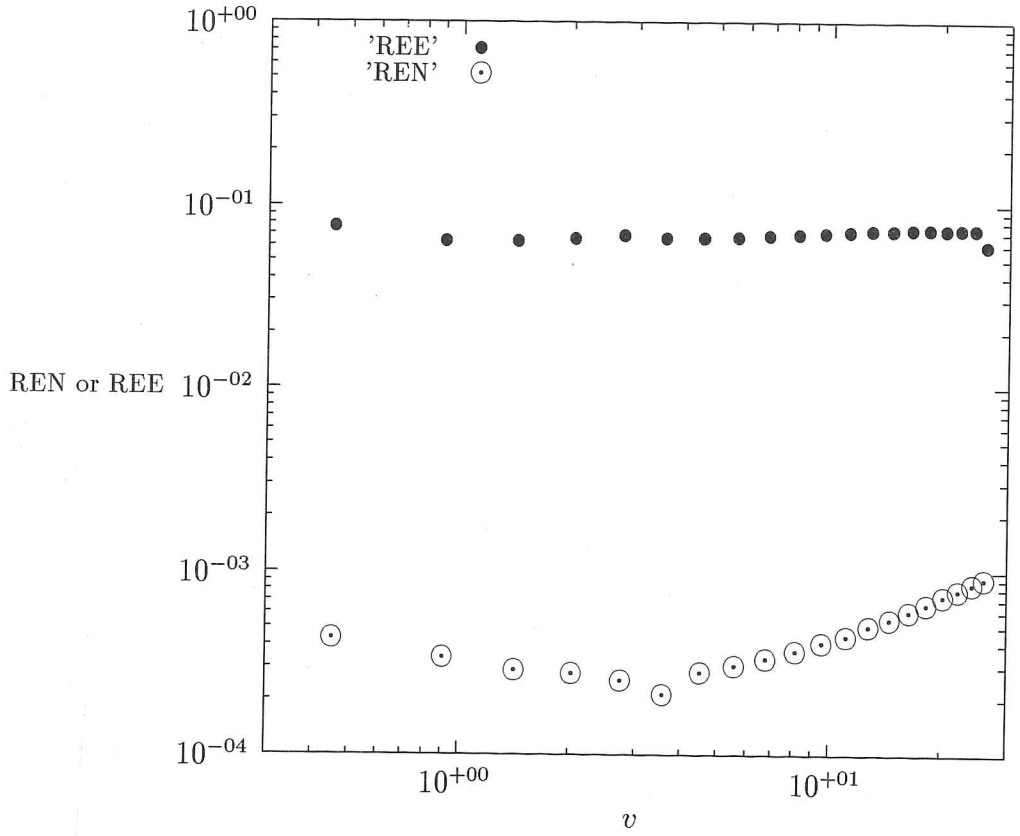

Figure 4.35: REN and REE values over the automatically refined discretization for the combined aggregation, growth and nucleation problem 4a

The REE and REN values of each element of the automatically refined domain were calculated and are plotted in figure 4.35. It can be seen that after the automatic refinement procedure has been performed, estimate (†) is still capable of predicting upper bounds of the error in the numerically obtained solutions (*ie.* the REE values are greater than their corresponding REN values).

As in the aggregation problem a large number of simulations were performed to compare the convergence properties of the finite element method over geometrically and automatically discretized domains. In the case of geometrically discretized domains the first element length was held constant at $h_1 = 0.5$ and the number of elements was increased according to a geometric progression from 10 to 100 elements. While in the

case of automatic discretization, refinement of a coarse geometric discretization of 10 elements was performed. A different Δ value was used in each simulation. This value was adjusted so that final discretizations consisted of between 10 and 129 elements after one application of the automatic refinement algorithm. In each case the rL_2 value and the CPU time required for convergence were recorded.

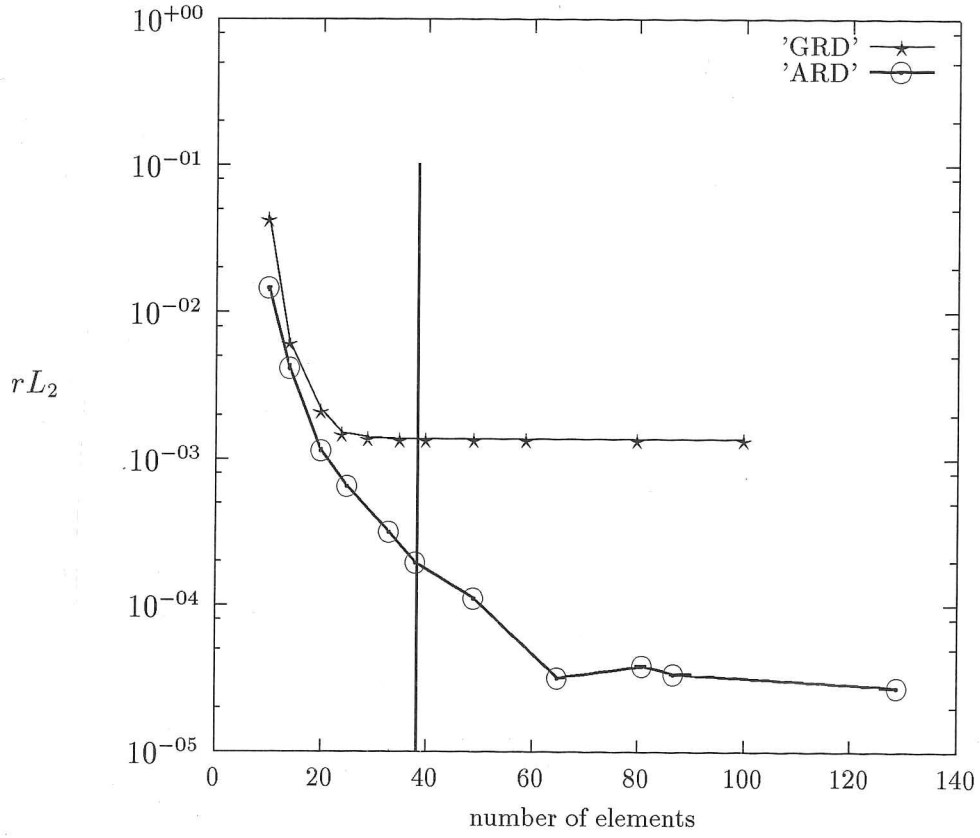

Figure 4.36: A comparison of the rL_2 values as functions of the number of elements spanning the domain for the combined aggregation, growth and nucleation problem over geometrically refined domains (GRD) and automatically refined domains (ARD).

Figure 4.36 shows rL_2 values as a function of the number of elements comprising a domain for several simulations over geometrically and automatically refined domains. The rL_2 values obtained over geometrically refined domains (GRD) are shown as stars ($\star$) while those obtained over automatically refined domains are shown as

odots ($\odot$). In all cases, solutions obtained over ARDs were of significantly higher quality (*ie.* had significantly lower rL_2 values than those obtained over GRDs composed of the same number of elements. The vertical line corresponds to the number of elements comprising the automatically refined domain when an error tolerance of $\Delta = 0.05$ is used. The practical implications of this line will be discussed in subsection 4.8.7.

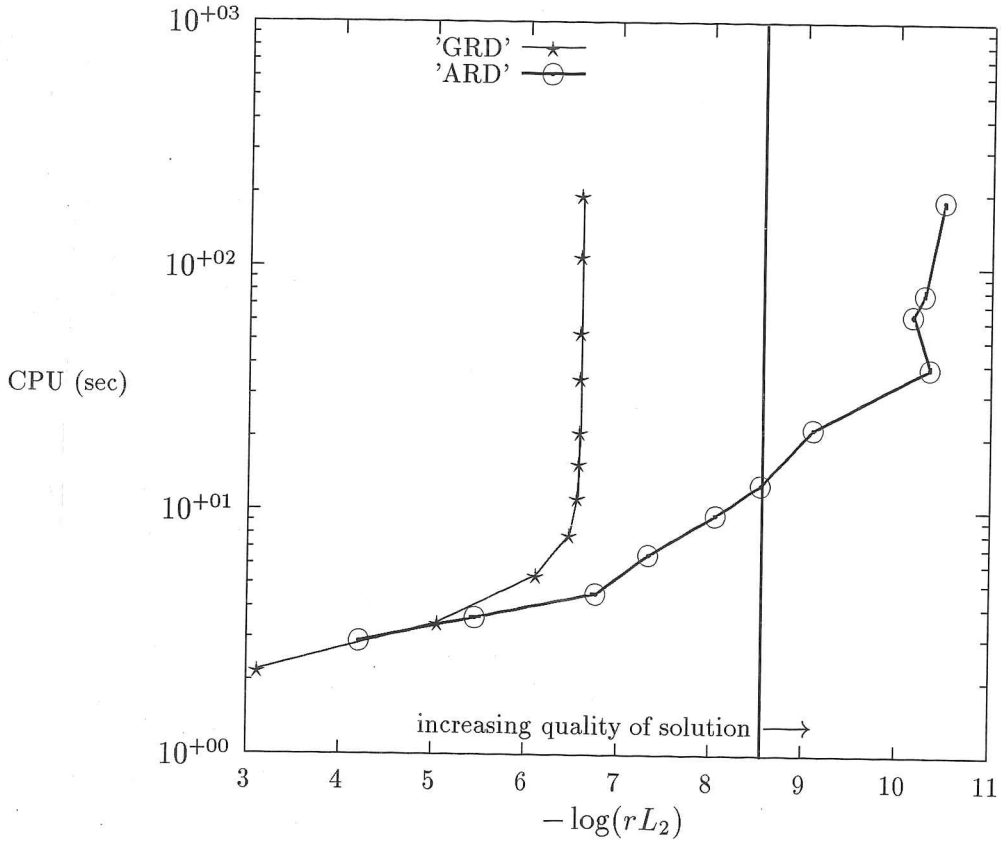

Figure 4.37: A comparison of the computation cost of a solution to the combined aggregation, growth and nucleation problem as a function of its quality over geometrically refined domains (GRD) and automatically refined domains (ARD).

In figure 4.37 the cost of obtaining a numerical solution is plotted as a function of is quality. The quality of a numerical solution is defined as in the previous case study while the cost is the CPU requirements in seconds. Points corresponding to solutions

over geometrically refined domains are once again represented by stars ($\star$) while those over automatically refined domains are represented by odots ($\odot$). These simulations indicate that high quality solutions may be obtained over ARDs at a fraction of the cost that would be required if the same quality solutions was to be obtained over a GRD. The vertical line corresponds to the quality of solution obtained over an automatically refined domain when an error tolerance of $\Delta = 0.05$ is used.

4.8 Discussion

4.8.1 Automation of the Refinement Procedure

The refinement procedure described in section 4.5 is automated by attempting to equilibrate the REE within each element. This was successfully achieved in the problems with aggregation (see figures 4.1-4.3), breakage (see figures 4.9-4.10) and combined aggregation, growth and nucleation (see figures 4.29-4.32). In each of these cases only two or three applications of the refinement algorithm were required to reduce the REE of each element to below the pre-specified error tolerance (Δ).

Automation of the refinement procedure proved more difficult for the growth problem. Some REE values remained greater than the pre-specified error tolerance after several iterations of the refinement procedure (see figures 4.19-4.22). These difficulties can be mainly attributed to the fact that the growth problem is linear while the error estimate ($\dagger$) has been derived for a non-linear equation. Hence the REN values display different trends to the REN values (see figure 3.3).

In all cases the automatic refinement algorithm added slightly to the cost of obtaining a solution however the quality of the solution was significantly improved by the automatic refinement.

4.8.2 Characteristics of Geometrically and Automatically Refined Discretizations

General trends in the GRDs and ARDs can be observed in all cases where the solution to the problem is monotonically decreasing (ie. aggregation, breakage and combined aggregation, growth and nucleation problems) :

- Apart from the first element, the ARD used fewer, larger elements to span the lower volume ranges and more, smaller elements to span the upper volume ranges (see figures 4.4, 4.13 and 4.33).
- Excluding the first element, element lengths increased monotonically as larger size ranges were reached.

This second trend does not occur in the ARD for the growth problem where the largest elements are located in the vicinity of the mode of the solution (see figure 4.24).

Hence in cases where the solution is monotonically decreasing it would be possible to select the parameters of the geometric progression so that the GRD and ARD would produce very similar results. This occurs in some of the simulations of the breakage problem and is discussed in further detail in subsection 4.8.5.

In the case of solutions with a mode the GRD and ARD will always result in very different discretizations of the domain.

4.8.3 Equilibration of the REN Values

In section 4.4 it was declared that the goal of the automatic refinement procedure was to improve the overall quality of the solution by equilibrating the REN values in each element. It was hoped that this could be indirectly achieved through equilibration

of the REE values. Figures 4.5, 4.14, 4.25 and 4.34 show the REN values over the initial geometrically refined domain and over the final automatically refined domains. In all of these figures (with the possible exception of figure 4.14) the spread of REN values appears to be equally broad regardless of the refinement procedure employed. Rather than attempting to quantify the spread of REN values (which would also have to be weighted somehow by element lengths) as a measure of the overall quality of a numerical solution the rL_2 values will be used for this purpose in subsection 4.8.5

4.8.4 REE and REN Values After Automatic Refinement

In figures 4.6, 4.15, 4.26 and 4.35 it can be seen that the REE values still bound the REN values from above in all cases. However, the REE values do not predict correct trends in the behaviour of the REN values as they did over geometrically refined domains. In the aggregation, growth and combined aggregation, growth and nucleation problems the REE values overestimate the REN values by smaller and smaller factors as large volumes are reached; the opposite trend is observed in the breakage problem.

4.8.5 Comparison of rL_2 Values Over GRDs and ARDs

In figures 4.7, 4.16, 4.27 and 4.36 it can be seen almost without exception lower rL_2 values are obtained over ARDs than GRDs using the same number of elements. Over very sparse discretizations of the aggregation problem the GRDs produced lower rL_2 values in a few simulations (see figure 4.7). The only other observations of rL_2 values of similar magnitude were made over very fine discretizations of the breakage problem. In this case the rL_2 values appeared to be converging to similar values (see figure 4.16). Close scrutiny revealed that for these simulations similar discretizations were being obtained by both refinement procedures except in the tail region where some elements of the ARD where about half the length of those of the GRD.

In general the rL_2 values over ARDs were about an order of magnitude lower than those over GRDs using the same number of elements.

4.8.6 Comparison of CPU Times Over GRDs and ARDs

Figures 4.8, 4.17, 4.28 and 4.37 are plots of the CPU requirements of solving a problem over either a GRD or and ARD as a function of the accuracy of the solution obtained. Solutions over the ARD are obtained in a fraction of the time required over a GRD in all cases where a solution quality $-\log(rL_2) > 5$ was achieved. A few simulations (over sparse discretizations) occurred where GRDs achieved better solutions using less CPU time. In all of these examples the ARD first converged to a solution over a sparse discretization and then refined it to an even sparser discretization. In the aggregation, growth and combined aggregation, growth and nucleation problems, solutions over GRDs reached limiting $-\log(rL_2)$ values of about 6 in the first two cases and 6.5 in the last case (see figures 4.8, 4.28 and 4.37) where further increases in CPU time resulting in negligible improvements in the solution quality. The limiting $-\log(rL_2)$ values over the ARDs were several units larger (from 8 to 10 units).

4.8.7 Practical Implementation of the Automatic Refinement Procedure

The application of the automatic refinement algorithm would be impractical if an entire series of simulations needed to be performed to solve each problem (as was done in each of the case studies of this chapter). In fact, much of this process was unnecessary since the refinement process is based upon an error estimate. It has been demonstrated that if a maximum acceptable error tolerance Δ is specified then a new discretization will be found such that this tolerance is not exceeded in any region of the domain ie. the REN values will be lower than Δ in each element. Admittedly

in figures 4.6, 4.15, 4.26 and 4.35 it can be seen that considerable uncertainty exists with regards to how much lower each of the REN values will be than Δ at various points along the domain.

In figures 4.7, 4.16, 4.27 and 4.36 a vertical line has been drawn corresponding to the number of elements resulting from an automatic discretization using an error tolerance of $\Delta = 0.05$. In each case a lower rL_2 value has been obtained over the ARD than for a GRD consisting of the same number of elements. In figures 4.8, 4.17, 4.28 and 4.37 this vertical line has been drawn to correspond to the rL_2 values obtained over an ARD using $\Delta = 0.05$ as an error tolerance. In the aggregation, growth and combined aggregation, growth and nucleation problem the vertical line does not intersect at any place with the CPU vs rL_2 curves for the GRDs. In these cases rL_2 values obtained over ARDs using error tolerances of $\Delta = 0.05$ could not be obtained over GRDs using reasonable amounts of CPU.

In all cases high quality solutions ($-\log(rL_2) > 6.5$) were obtained over the ARDs when the error tolerance was set to $\Delta = 0.05$.

4.9 Chapter Conclusions

The following conclusions can be made with regards to the automatic refinement algorithm described in this chapter :

- The ARD successfully automates the refinement procedure (except in growth only problems due to a poor correlation in the trends of REE and REN values). For aggregation, breakage, and combined aggregation, growth and nucleation problems the refinement procedure is usually completed in two or three applications of the algorithm. The user needs only to specify a maximum permissible error tolerance and a new discretization is found such that this error tolerance is not exceeded in any element of the domain.

- The refinement procedure did not affect the ability of error estimates (†) and (‡) to bound the error in the numerical solution.

- In all the investigated case studies, solutions obtained over ARDs were of higher quality than could be obtained over a GRD using the same number of elements. A few exceptions to this statement occurred in cases where low quality solutions were obtained over very sparse discretization.

- In all the investigated case studies, solutions were obtained over ARDs using much less CPU time than those of the same quality over GRDs.

Chapter 5

A Finite Element Method for Solving the Dynamic Population Balance Equation

In this chapter a finite element method is proposed for solving the dynamic population balance equation. After the derivation of the method, discussion is presented on several vital issues concerning appropriate specification of the domain. The method is used to solve several dynamic cases of the PBE and these results are compared favourably with those obtained using other methods.

5.1 Statement of the Problem

In the case of the dynamic population balance equation the problem becomes one of finding a function $n(v,t)$ (and the moments of this function) such that :

$$\begin{aligned}\frac{\partial n(v,t)}{\partial t} + \frac{\partial (G(v,t)n(v,t))}{\partial v} = & \frac{1}{2}\int_0^v \beta(v-w,w,t)n(v-w,t)n(w,t)dw \\ & - n(v,t)\int_0^\infty \beta(v,w,t)n(w,t)dw \\ & + \int_v^\infty \rho(v,w,t)S(w,t)n(w,t)dw \\ & - S(v,t)n(v,t) \qquad \forall\, v,t \,\epsilon\, (0,\infty]\end{aligned}$$

with the initial condition :

$$n(v,0) = n_{in}(v) \qquad \forall\, v \,\epsilon\, (0,\infty]$$

and subject to the dynamic boundary condition :

$$n(0,t) = n_0(t) \tag{5.1}$$

In this case the moments of the density distribution (for which we also wish to solve) are dynamic quantities which may be defined :

$$m_i(t) = \int_0^\infty v^i n(v,t)dv \tag{5.2}$$

5.2 Objectives of this Chapter

The work contained in this chapter has been performed with three main objectives in mind :

- To develop a method capable of solving the dynamic population balance equation for the full range of particulate behaviour : aggregation, breakage, growth and nucleation. Such a method is yet to be developed. Existing methods

tend to concentrate on one (Eyre, Wright and Reuter (1988), Steemson and White (1988), Gelbard, Tambour, and Seinfeld (1980), Hill and Ng (1995 and 96), Kim, and Tarbell (1990)), two (Kumar and Ramkrishna (1996a and b), Pilinis (1990), Warren and Seinfeld (1985)) or at most three (Gelbard and Seinfeld (1978), Viljoen, Eyre and Wright (1990), Eyre, Wright and Reuter (1988), Erasmus, Eyre and Everson (1994), Litster, Smit and Hounslow (1995)) of the above mentioned phenomena.

- To highlight several issues that will be inevitably encountered when attempting to use a discrete method to solve the dynamic PBE. Existing work frequently neglects to consider domain issues such as : where to truncate the domain, how to discretize the domain and how to select an appropriate size for each time step. In this chapter it will be demonstrated that great care must be taken with each of these items or else the method used to solve the problem can become inaccurate and/or unstable.

- The final objective is to attempt to resolve some of the above-mentioned domain issues. At this stage it is not possible to employ the same rigourous approach to domain discretization and refinement as was used for the steady state population balance equation. The theory of nonlinear error estimates for dynamic problems is still in its infancy. At the time of writing no useful results could be found that would lead to the derivation of an error estimate in terms of calculable parameters. Hence in this chapter the need for such developments is acknowledged and a finite element method is derived in such a way that future developments may be incorporated into an error estimate and automatic refinement scheme similar to those derived in chapters 3 and 4. In the numerical case studies of section 5.13 heuristic approaches to domain discretization and refinement are used in the absence of better methods.

5.3 Derivation of the Method

The derivation of a finite element method for the dynamic PBE will essentially follow the same sequence of steps described in section 2.3. However, a different approximation will be made on the form of the solution to the problem. This will reduce the problem to a system of first order ordinary differential equations that may be solved using standard time integration methods.

The finite element method for the dynamic PBE will be derived over an unscaled domain. As in the steady state case domain scaling will here be avoided since there is no systematic means of selecting scaling parameters (which would need to be time dependent) and since calculation of the moments of a distribution is much more difficult over a scaled domain. The method will however be derived over a general discretization of the domain to assist the incorporation of automatic refinement algorithms should they be developed in future work.

5.3.1 Approximation of the Solution

In dynamic formulations the solution to a problem in the e^{th} element at time t is approximated as follows :

$$n(v,t) \approx n_h^e(v,t) = \sum_{j=1}^{p} n_j^e(t)\psi_j^e(v) \qquad \forall\, v \,\epsilon\, \Omega_e \tag{5.3}$$

This is essentially the same approximation as that made for the steady state problem however in this case the nodal values of the solution have been assumed to be time dependent. Once again the Lagrange polynomials have been used for a basis of interpolation functions ($\psi_j^e(v)$).

5.3 Derivation of the Method

The derivation of a finite element method for the dynamic PBE will essentially follow the same sequence of steps described in section 2.3. However, a different approximation will be made on the form of the solution to the problem. This will reduce the problem to a system of first order ordinary differential equations that may be solved using standard time integration methods.

The finite element method for the dynamic PBE will be derived over an unscaled domain. As in the steady state case domain scaling will here be avoided since there is no systematic means of selecting scaling parameters (which would need to be time dependent) and since calculation of the moments of a distribution is much more difficult over a scaled domain. The method will however be derived over a general discretization of the domain to assist the incorporation of automatic refinement algorithms should they be developed in future work.

5.3.1 Approximation of the Solution

In dynamic formulations the solution to a problem in the e^{th} element at time t is approximated as follows :

$$n(v,t) \approx n_h^e(v,t) = \sum_{j=1}^{p} n_j^e(t)\psi_j^e(v) \qquad \forall\, v \,\epsilon\, \Omega_e \tag{5.3}$$

This is essentially the same approximation as that made for the steady state problem however in this case the nodal values of the solution have been assumed to be time dependent. Once again the Lagrange polynomials have been used for a basis of interpolation functions ($\psi_j^e(v)$).

5.4 Formation of a Weighted Residual Expression

Equation (1.7) is reformulated as a weighted residual statement in order to generate a system of ordinary differential equations.

Equation (1.7) may be re-arranged to obtain the following form :

$$\frac{\partial n(v,t)}{\partial t} + r[v,n,t]n(v,t) + G(v,t)\frac{\partial n(v,t)}{\partial v} = b^a[v,n,t] + b^b[v,n,t] \qquad (5.4)$$

where :

$$r[v,n,t] = \frac{\partial G(v,t)}{\partial v} + S(v,t) + \int_0^\infty \beta(v,w,t)n(w,t)dw \qquad (5.5)$$

and the birth rate due to aggregation ($b^a[v,n,t]$) and the birth rate due to breakage ($b^a[v,n,t]$) are as defined in expressions (1.3) and (1.5) respectively.

A weighted residual statement is formed from expression (5.5) by multiplying by a weight function $\phi(v)$ and integrating over the domain of element e :

$$\int_{v_a^e}^{v_b^e} \phi(v) \left\{ \frac{\partial n(v,t)}{\partial t} + r[v,n,t]n(v,t) + G(v,t)\frac{\partial n(v,t)}{\partial v} \right\} dv = \int_{v_a^e}^{v_b^e} \phi(v) \left\{ b^a[v,n,t] + b^b[v,n,t] \right\} dv \qquad (5.6)$$

The dynamic finite element approximation of the solution (5.3) is substituted into the above equation :

$$\sum_{j=1}^{p} \int_{v_a^e}^{v_b^e} \phi(v) \left\{ \dot{n}_j^e(t)\psi_j^e(v) + \left(r[v,n,t]\psi_j^e(v) + G(v,t)\frac{d\psi_j^e(v)}{dv} \right) n_j^e(t) \right\} dv = \int_{v_a^e}^{v_b^e} \phi(v) \left\{ b^a[v,n,t] + b^b[v,n,t] \right\} dv \qquad (5.7)$$

and expression (5.7) is used to generate a set of p equations by substitution of the set of p weight functions $\{\phi_i^e(v)\}_{i=1}^p$ for $\phi(v)$:

$$\sum_{j=1}^{p} \int_{v_a^e}^{v_b^e} \left\{ \phi_i^e(v)\psi_j^e(v) \right\} dv \; . \; \dot{n}_j^e(t)$$
$$+ \sum_{j=1}^{p} \int_{v_a^e}^{v_b^e} r[v,n,t]\phi_i^e(v)\psi_j^e(v) + G(v,t)\phi_i^e(v)\frac{d\psi_j^e(v)}{dv} dv \; . \; n_j^e(t) =$$
$$\int_{v_a^e}^{v_b^e} \phi_i^e(v) \left\{ b^a[v,n,t] + b^b[v,n,t] \right\} dv \quad (5.8)$$

where $i = 1, 2, \ldots, p$. Or in matrix notation :

$$A_{ij}^e \dot{n}_j^e[t] + B_{ij}^e[\boldsymbol{n},t] n_j^e[t] = F_i^e[\boldsymbol{n},t] \quad (5.9)$$

where :

$$A_{ij}^e = \int_{v_a^e}^{v_b^e} \phi_i^e(v)\psi_j^e(v) dv \quad (5.10)$$

$$B_{ij}^e = \int_{v_a^e}^{v_b^e} r[v,n,t]\phi_i^e(v)\psi_j^e(v) + G(v,t)\phi_i^e(v)\frac{d\psi_j^e(v)}{dv} \; dv \quad (5.11)$$

and :

$$F_i^e = \int_{v_a^e}^{v_b^e} \phi_i^e(v) \left\{ b^a[v,n,t] + b^b[v,n,t] \right\} dv \quad (5.12)$$

The argument $(\boldsymbol{n})$ in equation (5.9) is used to denote that the elements of the B-matrix and the F-vector are dependent upon the vector of unknown nodal values.

5.5 Weight Functions

As in the steady state case different choices of weight functions result in different finite element formulations. The weight functions giving rise to the collocation and the Galerkin formulations will again be discussed.

5.5.1 The Collocation Formulation

If the set of Dirac delta functions (2.20) are used as weight functions a collocation formulation of the method results. In this case the A-matrix, the B-matrix and the F-vector of equation (5.9) become :

$$A^e_{ij} = \psi^e_j(v^e_i) \tag{5.13}$$

$$B^e_{ij} = r[v^e_i, n, t]\psi^e_j(v^e_i) + G(v^e_i, t)\frac{d\psi^e_j(v^e_i)}{dv} \tag{5.14}$$

and :

$$F^e_i = b^a[v^e_i, n, t] + b^b[v^e_i, n, t] \tag{5.15}$$

5.5.2 The Galerkin Formulation

In the Galerkin formulation the Lagrange interpolation polynomials are also used as the weight functions (see expression (2.26)). With this choice of weight functions the A-matrix, the B-matrix and the F-vector become :

$$A^e_{ij} = \int_{v^e_a}^{v^e_b} \psi^e_i(v)\psi^e_j(v)dv \tag{5.16}$$

$$B^e_{ij} = \int_{v^e_a}^{v^e_b} r[v, n, t]\psi^e_i(v)\psi^e_j(v) + G(v, t)\psi^e_i(v)\frac{d\psi^e_j(v)}{dv}\, dv \tag{5.17}$$

and :

$$F^e_i = \int_{v^e_a}^{v^e_b} \psi^e_i(v)\left\{b^a[v, n, t] + b^b[v, n, t]\right\} dv \tag{5.18}$$

The Galerkin formulation is particularly relevant in dynamic problems since it permits the boundary condition to be implemented in a straight forward manner. Boundary conditions will be discussed in section 5.7.

5.6 Nodal Approximations of the Birth and Death Terms

The elements of the B-matrix and F-vector are operators of the density distribution and must be reformulated in terms of nodal values. These terms may be approximated in the same way as for the steady state problem since each of the parametrizing functions ($\beta(v,w,t), \rho(v,w,t), S(v,t)$ and $G(v,t)$) are assumed to have known time dependencies.

5.7 Boundary Conditions

A slight modification is made to the derivations of sections 5.5.1 and 5.5.2 in order to implement the boundary condition $n(0,t) = n_0(t)$. Consider equation (5.6) :

$$\int_{v_a^e}^{v_b^e} \phi(v) \left\{ \frac{\partial n(v,t)}{\partial t} + r[v,n,t]n(v,t) + G(v,t)\frac{\partial n(v,t)}{\partial v} \right\} dv = \int_{v_a^e}^{v_b^e} \phi(v) \left\{ b^a[v,n,t] + b^b[v,n,t] \right\} dv \quad (5.19)$$

If the flux term is integrated by parts then this equation becomes :

$$\int_{v_a^e}^{v_b^e} \left\{ \phi(v) \left(\frac{\partial n(v,t)}{\partial t} + r[v,n,t]n(v,t) \right) - \frac{\partial}{\partial v} \left(\phi(v)G(v,t) \right) n(v,t) \right\} dv + \left[\phi(v)G(v,t)n(v,t) \right]_{v_a^e}^{v_b^e} = \int_{v_a^e}^{v_b^e} \phi(v) \left\{ b^a[v,n,t] + b^b[v,n,t] \right\} dv \quad (5.20)$$

If we assume that the boundary condition will be applied at the point $v = 0$ and let the first element span the domain $v \in (0, v_b^1)$ then the above expression becomes :

$$\int_0^{v_b^1} \left\{ \phi(v) \left(\frac{\partial n(v,t)}{\partial t} + r[v,n,t]n(v,t) \right) - \frac{\partial}{\partial v} \left(\phi(v)G(v,t) \right) n(v,t) \right\} dv + \phi(v_b^1)G(v_b^1,t)n(v_b^1,t) - \phi(0)G(0,t)n(0,t) = \int_0^{v_b^1} \phi(v) \left\{ b^a[v,n,t] + b^b[v,n,t] \right\} dv \quad (5.21)$$

Then the boundary condition (5.1) may be neatly incorporated as follows :

$$\int_0^{v_b^1} \left\{ \phi(v) \left(\frac{\partial n(v,t)}{\partial t} + r[v,n,t]n(v,t) \right) - \frac{\partial}{\partial v} \left(\phi(v)G(v,t) \right) n(v,t) \right\} dv$$
$$+ \phi(v_b^1)G(v_b^1,t)n(v_b^1,t) - \phi(0)G(0,t)n_0(t)$$
$$= \int_0^{v_b^1} \phi(v) \left\{ b^a[v,n,t] + b^b[v,n,t] \right\} dv \quad (5.22)$$

As described in section 5.3.1 the dynamic finite element approximation of the solution (5.3) is made and a basis of weight functions is substituted into the above equations. A set of equations of the form (5.9) results where the matrices and the vector are of the form :

$$A_{ij}^1 = \int_0^{v_b^1} \psi_i^1(v)\psi_j^1(v)dv \quad (5.23)$$

$$B_{ij}^1 = \int_0^{v_b^1} r[v,n,t]\psi_i^1(v)\psi_j^1(v) - \frac{\partial}{\partial v} \left(G(v,t)\psi_i^1(v) \right) \psi_j^1(v)dv$$
$$+ G(v_b^1,t)\psi_i^1(v_b^1)\psi_j^1(v_b^1) \quad (5.24)$$

and :

$$F_i^1 = \int_0^{v_b^1} \psi_i^1(v) \left\{ b^a[v,n,t] + b^b[v,n,t] \right\} dv + \psi_i^1(0)G(0,t)n_0(t) \quad (5.25)$$

In this way a Galerkin formulation may be conveniently used over the first element to implement a boundary condition.

5.8 Assembly of the Elements into a Global System

Equations of the form (5.9) are assembled into a global system as described in section 2.9 by imposing continuity of the solution at the inter-element boundaries. In this

way the dynamic PBE is reduced to a set of first order ordinary differential equations which may be written as :

$$\dot{n}[t]_j^g = \left[A_{ij}^g\right]^{-1} \left(F_i^g[\boldsymbol{n},t] - B_{ij}^g[\boldsymbol{n},t]n_j^g[t]\right) \tag{5.26}$$

The superscript ^{g} in the above equation signifies elements of the global system.

5.9 Time Integration

Standard time integration techniques may be used to solve the system of equations (5.26). In this section the suitability of several of these techniques will be discussed and their performances compared in sub-section 5.14.1.

5.9.1 θ-Integration Techniques

Finite difference schemes may be used to reformulate (5.26) as algebraic expressions at each consecutive time step. This group of methods are commonly known as the θ integration family (specific examples include the forward difference or Euler, Crank-Nicolson, Galerkin and backward difference schemes all of which are discussed by Reddy (1993)). These schemes have been applied to the dynamic PBE by several workers Erasmus, Eyre and Everson (1994), Eyre, Wright and Reuter (1988) and Viljoen, Eyre, and Wright (1990). In most applications the implicit methods (Crank-Nicolson, Galerkin and backward difference method) are used in preference to the explicit Euler scheme due to their superior stability characteristics.

The main disadvantage to these methods is that no efficient means of automatically controlling the step size has been proposed. Such automatic time step control can increase the efficiency of a method by several orders of magnitude. Nevertheless these methods may be worthy of re-investigation once a dynamic error estimate and subsequent criterion for automatic time step control have been developed.

5.9.2 Runge-Kutta Methods

These methods are popular due to their simplicity and robustness. Several workers have applied fourth order Runge-Kutta methods to the dynamic PBE : Hill and Ng (1995 and 96), Hounslow, Ryall and Marshall (1988), Litster, Smit and Hounslow (1995) and Steemson and White (1988) but once again the computational efficiency of these methods was limited by the use of constant size time steps.

In section 5.13 a Runge-Kutta-Fehlberg method will be used to integrate expression (5.26) in conjunction with the algorithm described by Press, Teukolsky, Vetterling and Flannery (1992) that automatically adjusts the size of each time step so that the local truncation error is kept below a pre-specified bound.

5.10 The Modified Midpoint Method

The modified midpoint method when used in conjunction with the Bulirsch-Stoer technique (Stoer and Bulirsch (1980)) was recommended by Press, Teukolsky, Vetterling and Flannery (1992) as being the best known way to obtain high-accuracy solutions to ordinary differential equations with minimal computational effort. In addition an efficient automatic time step control strategy has been proposed for this method.

5.11 Stiff Solvers

The Rosenbrock method (as described by Kaps and Rentrop (1979)) and the Gear method (Gear (1971)) are implicit versions of the Runge-Kutta and the multistep method respectively. These methods have been specially designed to solve stiff systems and are reported to be capable of drastically reducing the number of time steps

required to solve a problem. The main disadvantage of these methods is that they require a Jacobian matrix to be evaluated. This matrix is completely dense in aggregation and breakage problems and must be numerically computed which significantly increases the computational load of each time step.

5.12 Domain Issues

As in the steady state problem, careful consideration must be given to truncation and discretization of the domain if a reasonable approximation of problem (1.7) is to be obtained. In the dynamic case however, optimal truncation points and discretizations of the domain can be expected to vary with time.

5.12.1 Domain Truncation

Existing methodologies for solving the dynamic PBE give little consideration to the manner in which the domain is truncated. In all of the articles reviewed in section 1.7 : a truncation point was selected (often arbitrarily) and held constant throughout each simulation. This practice inevitably leads to problems. In the simulations performed by Gelbard and Seinfeld (1978) finite domain errors increasingly degraded the quality of the solutions as time progressed. A simple way to overcome this problem is to select a truncation point so large that finite domain errors are insignificant throughout the entire simulation. Unfortunately very stiff systems of equations will result quite early in such simulations since the magnitude of the density distribution in the tail region is likely to be hundreds of orders of magnitude smaller than that at other regions of the domain. This problem will be highlighted in section 5.14.3.

Both of these problems will be avoided in the current work by using a time dependent truncation point (ie. $v_{max} = v_{max}(t)$). At each time step a new truncation point

will be selected as described in section 2.11 for the steady state problem such that criterion (2.47) is approximately satisfied.

5.12.2 Extrapolation of the Solution

In aggregation/growth dominant problems v_{max} must increase with time if criterion (2.47) is to be satisfied throughout the simulation ie. $v_{max}(t_s) < v_{max}(t_{s+1})$ for $t_s < t_{s+1}$. In such problems an extrapolation of the solution will be required over the extended domain. For instance consider the task of solving an aggregation/growth dominant PBE for the density distribution at time $t = t_{s+1}$. The density distribution at the previous time step (t_s) will already be known over the domain $v \in (0, v_{max}(t_s)]$. Equation (5.9) and a time integration procedure may then be used to solve for the density distribution at time $t = t_{s+1}$ over the same domain. However, criterion (2.47) will not be satisfied over this domain unless a new, larger truncation point $v_{max}(t_{s+1})$ is found. Some expression for $n(v, t_s)$ must be available in the region $(v_{max}(t_s), v_{max}(t_{s+1})]$ if $n(v, t_{s+1})$ is to be evaluated whilst simultaneously satisfying criterion (2.47). Thus the need for extrapolation of $n(v, t_s)$.

In this thesis $n(v, t_s)$ will be extrapolated through the exponential function $\exp(a+bv)$ where the constants a and b are selected such that this exponential attains the nodal values at the first and last nodes of the final element of the discretization at time $t = t_s$. It is possible to use other extrapolation functions. Indeed a straight line could be used if the time steps were so small that $v_{max}(t_s)$ was not much larger than $v_{max}(t_{s+1})$. However, given the rapidly decreasing nature of most solutions to the PBE exponential functions are the simplest sensible choice.

5.12.3 Discretization of the Domain

In chapter 2 it was demonstrated that accurate solutions to the steady state PBE could be approached over domains that were discretized according to a geometric progression. The geometric discretization described in section 2.4 will also be used for the dynamic PBE until a more rigorous technique (such as the refinement technique described in chapter 4) is proposed.

Since the size of the domain changes at each time step a new discretization must also be defined at each time step. The heuristic to be used will scale each element of the old discretization by the factor $v_{max}(t_{s+1})/v_{max}(t_s)$ so that a new geometric discretization consisting of the same number of elements spans the new domain.

5.13 Numerical Case Studies

Four numerical case studies involving the various size enlargement phenomena will be performed to investigate whether the finite element method and time integration schemes are capable of predicting accurately the density distribution and moments for the dynamic PBE.

5.13.1 Case 1 : An Aggregation Problem

The dynamic PBE for aggregation is of the form :

$$\frac{\partial n(v,t)}{\partial t} = \int_0^{v/2} \beta(v-w,w,t)n(v-w,t)n(w,t)dw - n(v,t)\int_0^{\infty} \beta(v,w,t)n(w,t)dw \tag{5.27}$$

If the initial distribution is selected to be :

$$n_{in}(v) = \frac{N_0}{v_0}\exp(-v/v_0) \tag{5.28}$$

and the aggregation rate is assumed to be size and time independent :

$$\beta(v, w, t) = \beta_0 \tag{5.29}$$

then the analytical solution to this PBE has been derived by Scott (1968) as :

$$n(v, t) = \frac{4N_0}{v_0(\tau + 2)^2} \exp\left(\frac{-2v}{v_0(\tau + 2)}\right) \tag{5.30}$$

where : $\tau = N_0\beta_0 t$. Analytical expressions for the moments of this distribution are obtained by multiplying the PBE by v^i (where i is the relevant index) and integrating over the domain $v \in (0, \infty]$. Using this procedure the first three moments may be found to be :

$$m_0(t) = \frac{2N_0}{2 + \beta_0 N_0 t} \tag{5.31}$$

$$m_1(t) = N_0 v_0 \tag{5.32}$$

$$m_2(t) = N_0(2 + \beta_0 N_0 t)v_0^2 \tag{5.33}$$

This problem was investigated for the following set of parameters :

$$N_0 = v_0 = \beta_0 = 1 \tag{5.34}$$

over the initial domain $v \in (0, 11.3]$ which was spanned by 20 elements with the first element of length 0.5 units.

The Runge-Kutta-Fehlberg (RKF), the Bulirsch-Stoer (BS) and the Gear method were each used to integrate the system of equations (5.26) that resulted from the collocation formulation of the aggregation PBE. For each method records were made of the CPU time (in seconds) required by a Pentium 75 MHz computer to integrate the problem to a final value of $t = 1000$ units, the number of time steps required and the rL_2 value of the solution (as defined by expression (4.13)) at $t = 1000$. These values are shown in table 5.1.

Method	RKF	BS	Gear
CPU	22	29	63
steps	72	16	11
rL_2	0.73×10^{-2}	0.43×10^{-2}	1.42×10^{-2}

Table 5.1: CPU time, number of time steps required and the accuracy of the final solution when the Runge-Kutta-Fehlberg, the Bulirsch-Stoer and the Gear method were used to solve the dynamic aggregation problem to a time $t = 1000$.

The simulation was completed in the shortest amount of time by the Runge-Kutta-Fehlberg method despite the fact that this method required significantly more time steps. The Gear method proved to be the least accurate of the three methods with an rL_2 value greater than 10^{-2} while the Bulirsch-Stoer and Runge-Kutta-Fehlberg achieved more accurate solutions with rL_2 values of 0.73×10^{-2} and 0.43×10^{-2} respectively.

The finite element predictions of the density distribution are shown in figure 5.1 at three different times. The solid lines represent the analytical solution while the symbols ($\odot$) represent the FEM nodal values at the end-points of each element. These points were obtained using the Runge-Kutta-Fehlberg method to integrate the collocation formulation of equation (5.26). In each case the numerical predictions are in excellent agreement with the analytical solutions.

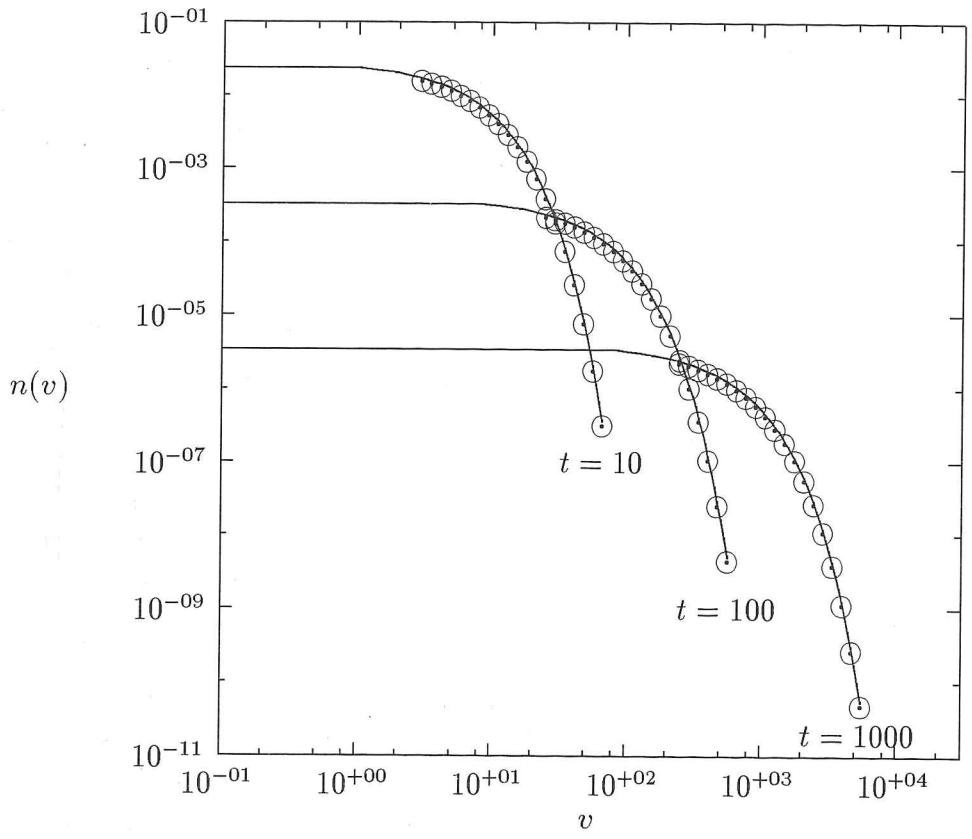

Figure 5.1: FEM predictions of the number density function ($n(v,t)$) for the dynamic aggregation problem where $\beta(v,w,t) = \beta_0 = 1$, $n_{in}(v) = \frac{N_0}{v_0}\exp(-v/v_0)$ and $N_0 = v_0 = 1$ at times $t = 10, 100$ and 1000.

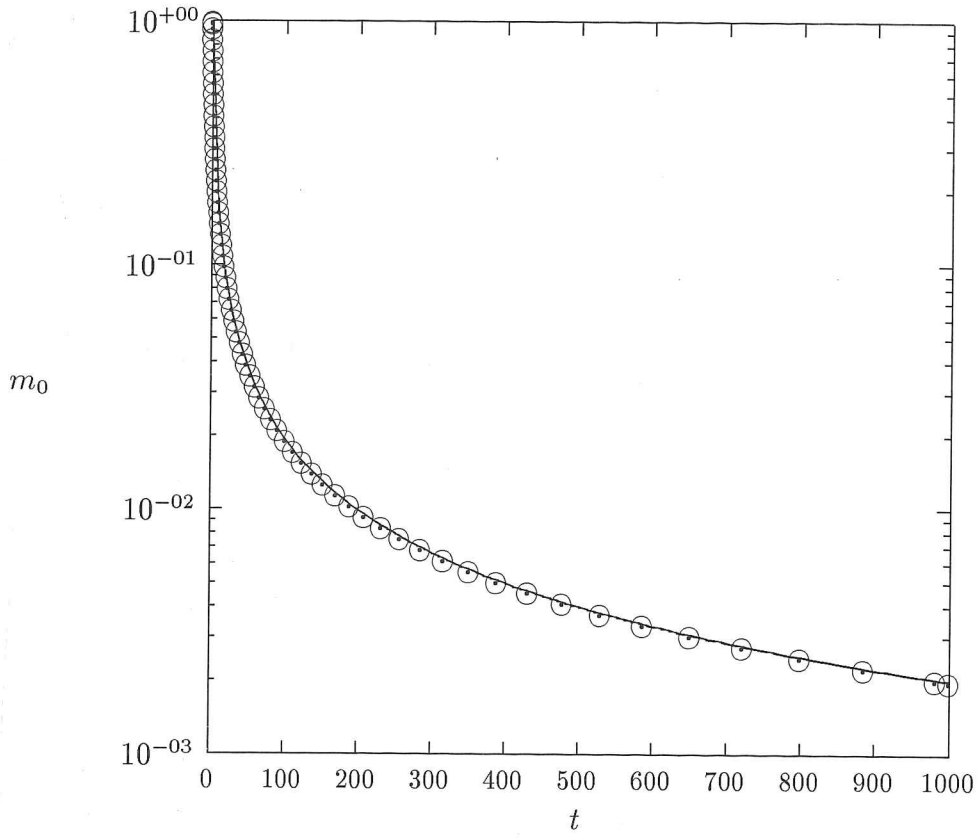

Figure 5.2: FEM predictions of the zeroth moment of the density distribution ($m_0(t)$) at each time step for the dynamic aggregation problem.

Numerical approximations of the moments of the density distribution were obtained at each time step (t_s) by multiplying the finite element solution by v^i (where $i = 0, 1, 2$ for the zeroth, first and second moments) and integrating over the domain $v \in (0, v_{max}(t_s)]$. The finite element predictions of the moments are shown in figures 5.2, 5.3 and 5.4 as symbols ($\odot$) at each Runge-Kutta-Fehlberg time step while the analytical expressions of the moments (5.31), (5.32) and (5.33) are plotted as solid lines. In each case the numerical predictions are in excellent agreement with the analytical expressions. The percentage errors in each of these moments at time $t = 1000$ was recorded in table 5.5 at the end of this section.

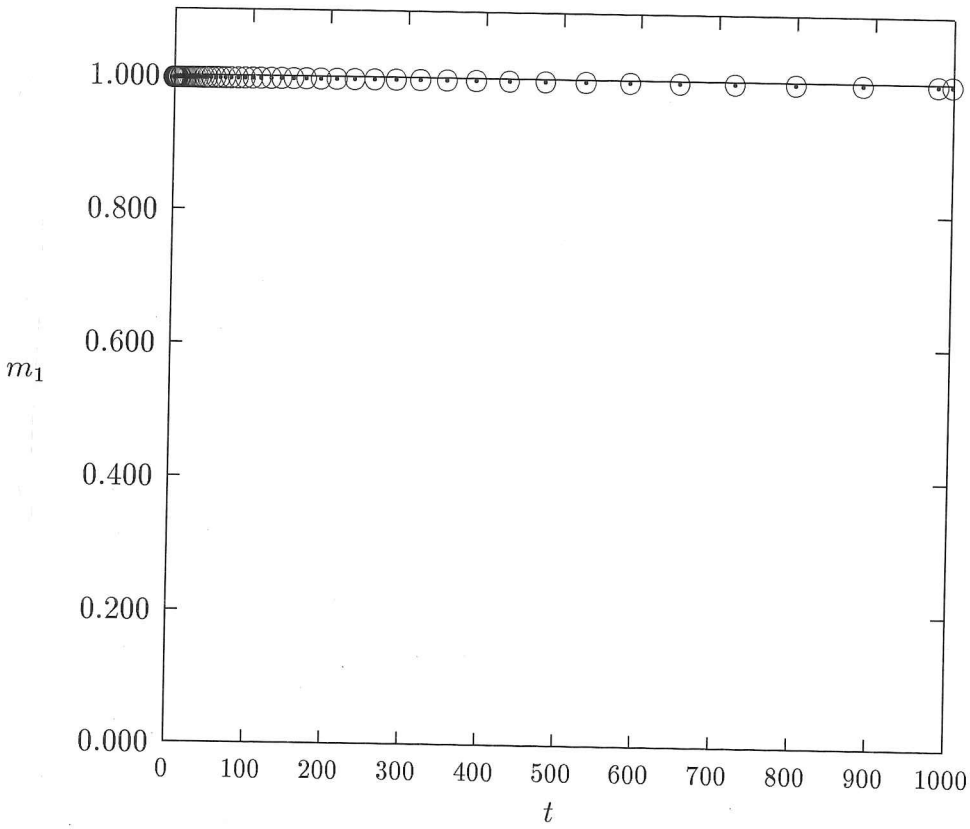

Figure 5.3: FEM predictions of the first moment of the density distribution ($m_1(t)$) at each time step for the dynamic aggregation problem.

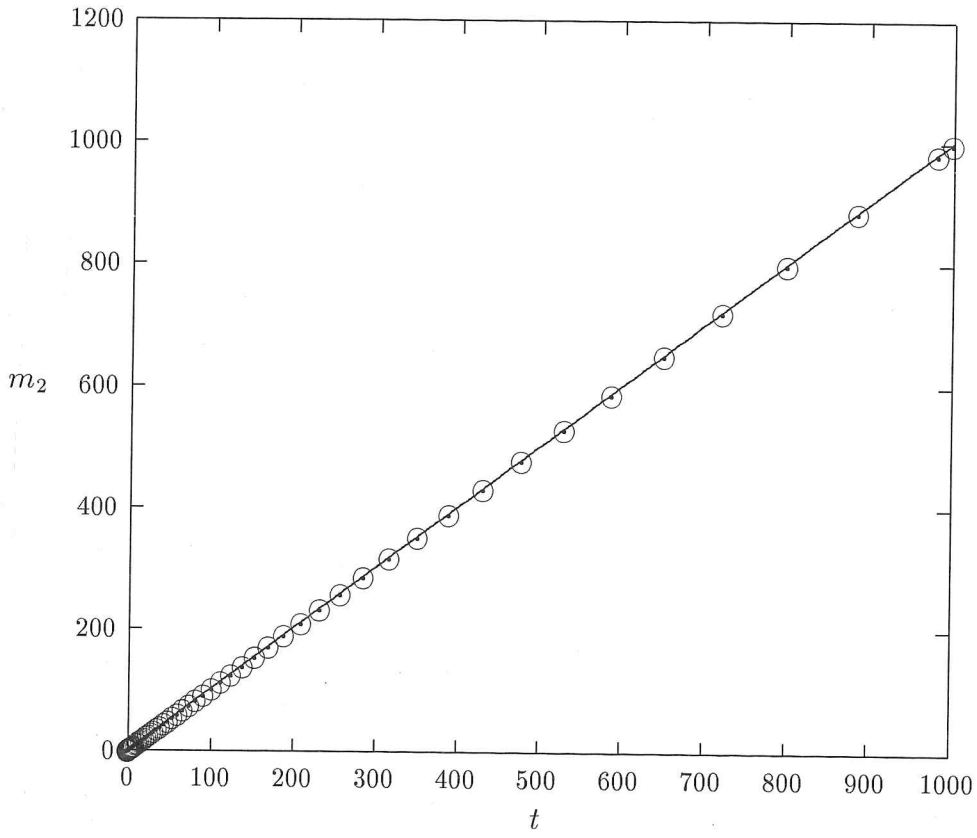

Figure 5.4: FEM predictions of the second moment of the density distribution ($m_2(t)$) at each time step for the dynamic aggregation problem.

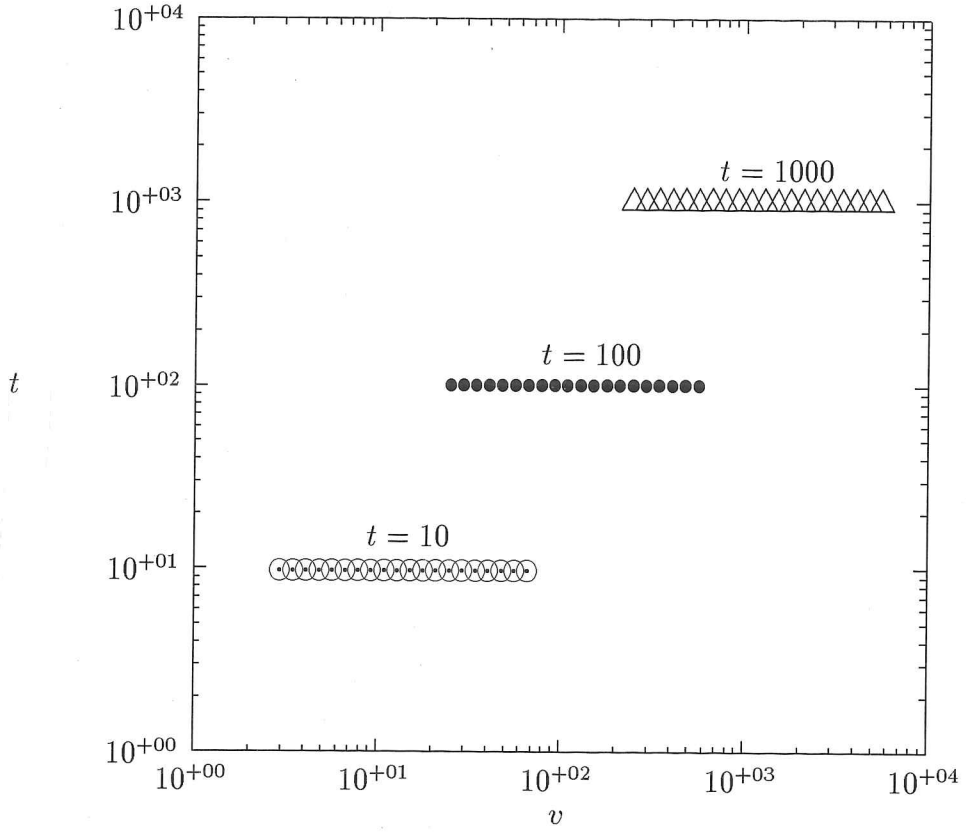

Figure 5.5: Locations of elements comprising the discretization over which the aggregation problem was solved at times $t = 10, 100$ and 1000.

The manner in which the discretization of the domain changed with time is illustrated in figure 5.5. This figure shows the locations of the end-points of each element comprising a discretization at the times $t = 10, 100$ and 1000. In each case the lower limit of the first element is located at $v = 0$, however this is not shown since a logarithmic scale is used for the volume axis. At each time step the finite domain is stretched by a constant factor as described in subsection 5.12.3.

5.13.2 Case 2 : A Breakage Problem

The dynamic PBE for breakage is of the form :

$$\frac{\partial n(v,t)}{\partial t} = \int_v^\infty \rho(v,w,t)S(w,t)n(w,t)dw - S(v,t)n(v,t) \tag{5.35}$$

If the initial distribution is selected to be :

$$n_{in}(v) = \frac{N_0}{v_0}\exp(-v/v_0) \tag{5.36}$$

and the breakage kernel and breakage functions are assumed to be time independent and of the following forms :

$$\rho(v,w,t) = \frac{2}{w} \qquad S(v,t) = v \tag{5.37}$$

then the analytical solution to this PBE has been derived by Ziff and McGrady (1991) as :

$$n(v,t) = \frac{N_0}{v_0}(1+tv_0)^2\exp(-(t+1/v_0)v) \tag{5.38}$$

Analytical expressions for the moments of this distribution are obtained by multiplying the PBE by v^i (where i is the relevant index) and integrating over the domain $v \epsilon (0,\infty]$. Using this procedure the first three moments may be determined to be :

$$m_0(t) = N_0(1+v_0t) \tag{5.39}$$

$$m_1(t) = N_0v_0 \tag{5.40}$$

$$m_2(t) = \frac{2N_0v_0^2}{1+v_0t} \tag{5.41}$$

This problem was investigated for the following set of parameters :

$$N_0 = v_0 = 1 \tag{5.42}$$

over the initial domain $v \epsilon (0, 11.3]$ which was spanned by 20 elements with the first element of length 0.5 units.

The Runge-Kutta-Fehlberg (RKF), the Bulirsch-Stoer (BS) and the Gear method were each used to integrate the system of equations (5.26) that resulted from the collocation formulation of the breakage PBE. In each case the CPU time (in seconds) required to integrate the problem to a final value of $t = 1000$ units, the number of time steps required and the rL_2 value of the solution at $t = 1000$ were recorded. These values are shown in table 5.2. The Runge-Kutta-Fehlberg method proved to be the fastest and most accurate method for solving this problem but once again it required significantly more time steps to reach a final time of $t = 1000$.

Method	RKF	BS	Gear
CPU	76	145	151
steps	207	50	13
rL_2	0.94×10^{-3}	0.19×10^{-2}	1.05×10^{-2}

Table 5.2: CPU time, the number of time steps required and the accuracy of the final solution when the Runge-Kutta-Fehlberg, the Bulirsch-Stoer and the Gear method were used to solve the dynamic breakage problem to a time $t = 1000$.

The finite element predictions of the density distribution are shown in figure 5.6 at three different times. The solid lines represent the analytical solution while the symbols ($\odot$) represent the FEM nodal values at the end-points of each element. These points were obtained using the Runge-Kutta-Fehlberg method to integrate the collocation formulation of equation (5.26). In each of these cases the numerical predictions are in excellent agreement with the analytical expressions.

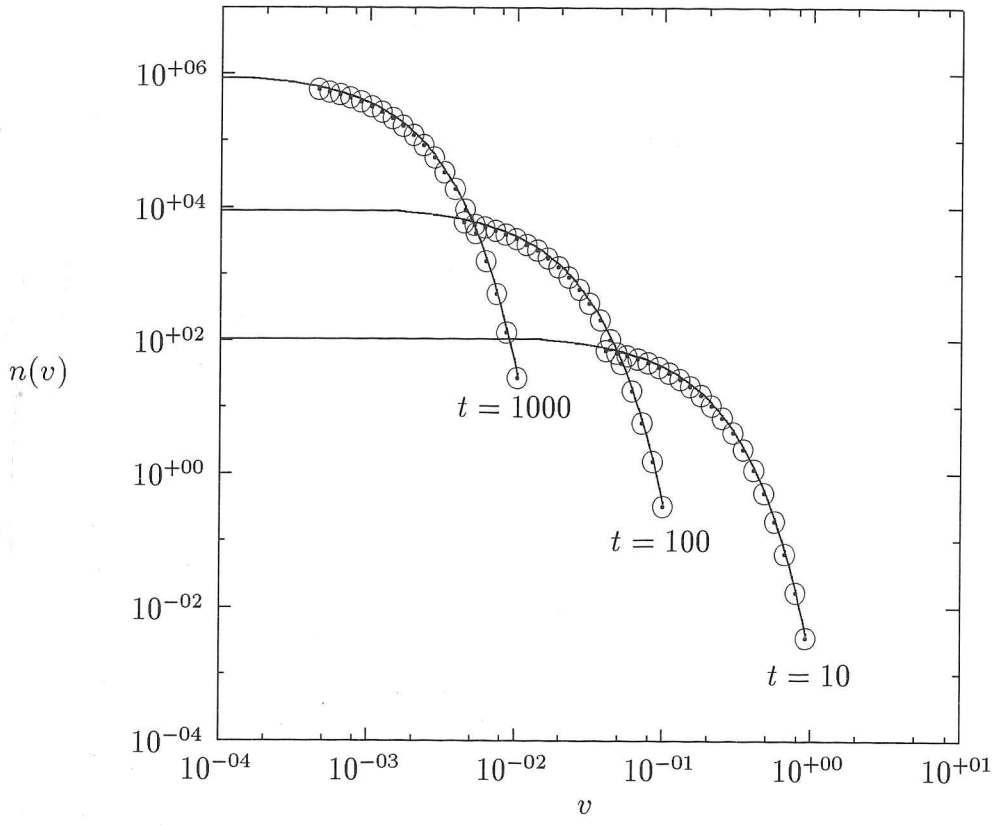

Figure 5.6: FEM predictions of the number density function $(n(v,t))$ for the dynamic breakage problem where $\rho(v,w,t) = 2/w$, $S(v,t) = v$, $n_{in}(v) = \frac{N_0}{v_0}\exp(-v/v_0)$ and $N_0 = v_0 = 1$ at times $t = 10, 100$ and 1000.

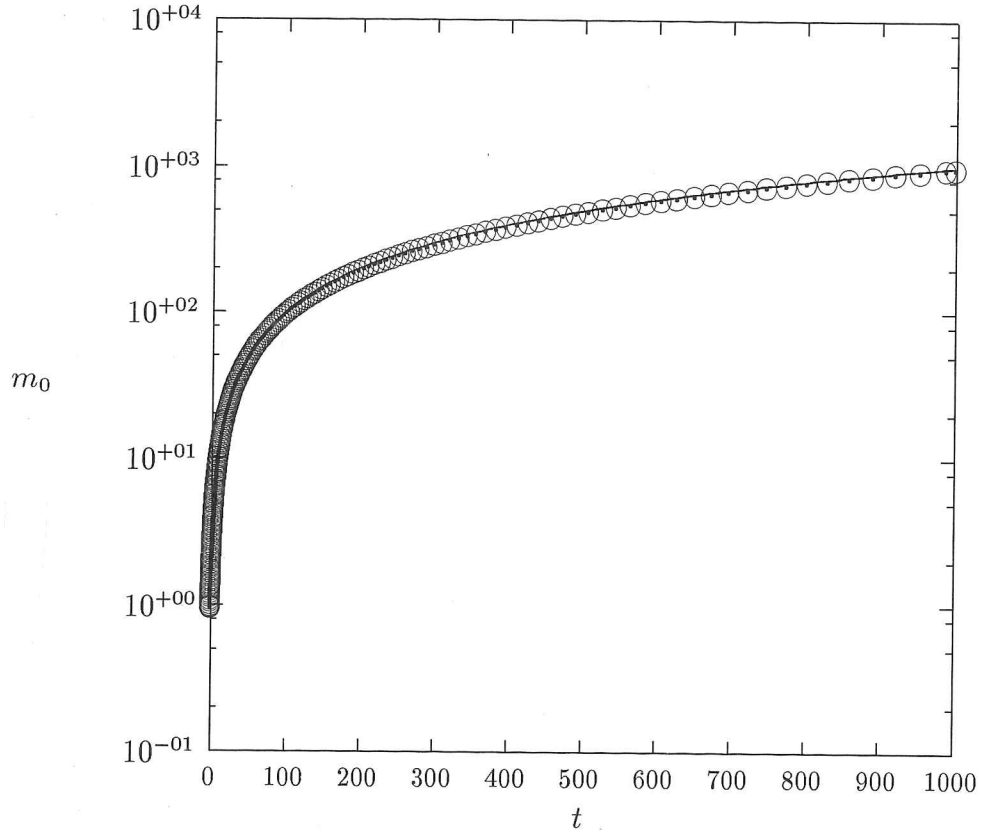

Figure 5.7: FEM predictions of the zeroth moment of the density distribution ($m_0(t)$) at each time step for the dynamic breakage problem.

Numerical approximations of the moments of the density distribution were obtained at each time step (t_s) by multiplying the finite element solution by v^i (where $i = 0, 1, 2$ for the zeroth, first and second moments) and integrating over the domain $v \in (0, v_{max}(t_s)]$. The finite element predictions of the moments are shown in figures 5.7, 5.8 and 5.9 as symbols ($\odot$) at each Runge-Kutta-Fehlberg time step while the analytical expressions for the moments (5.39), (5.40) and (5.41) are plotted as solid lines. In all cases the finite element predictions of the moments can be seen to be in excellent agreement with the analytical expressions. The percentage errors in each of these moments at time $t = 1000$ are recorded in table 5.5 at the end of this section.

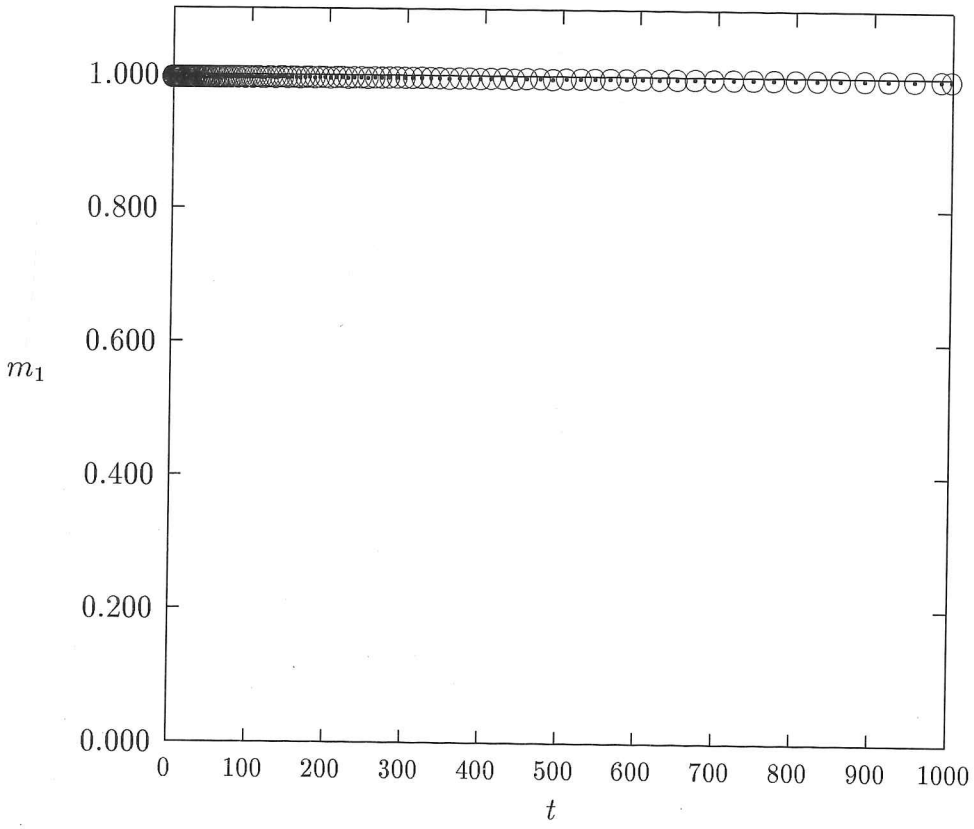

Figure 5.8: FEM predictions of the first moment of the density distribution ($m_1(t)$) at each time step for the dynamic breakage problem.

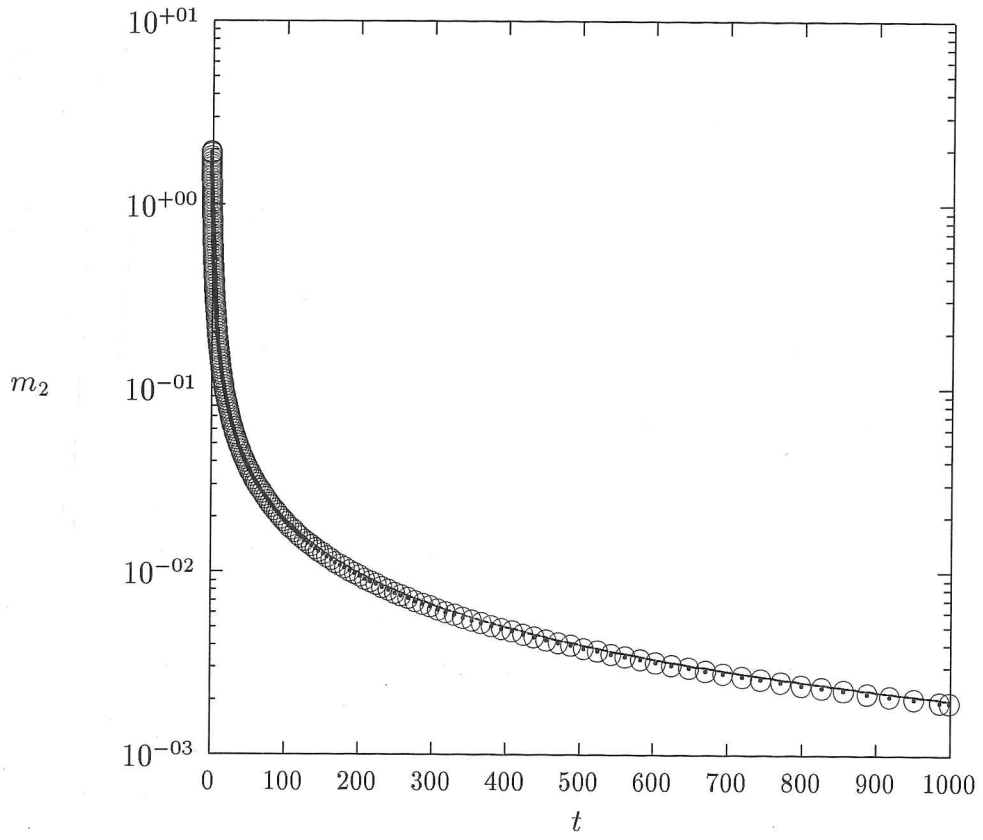

Figure 5.9: FEM predictions of the second moment of the density distribution ($m_2(t)$) at each time step for the dynamic breakage problem.

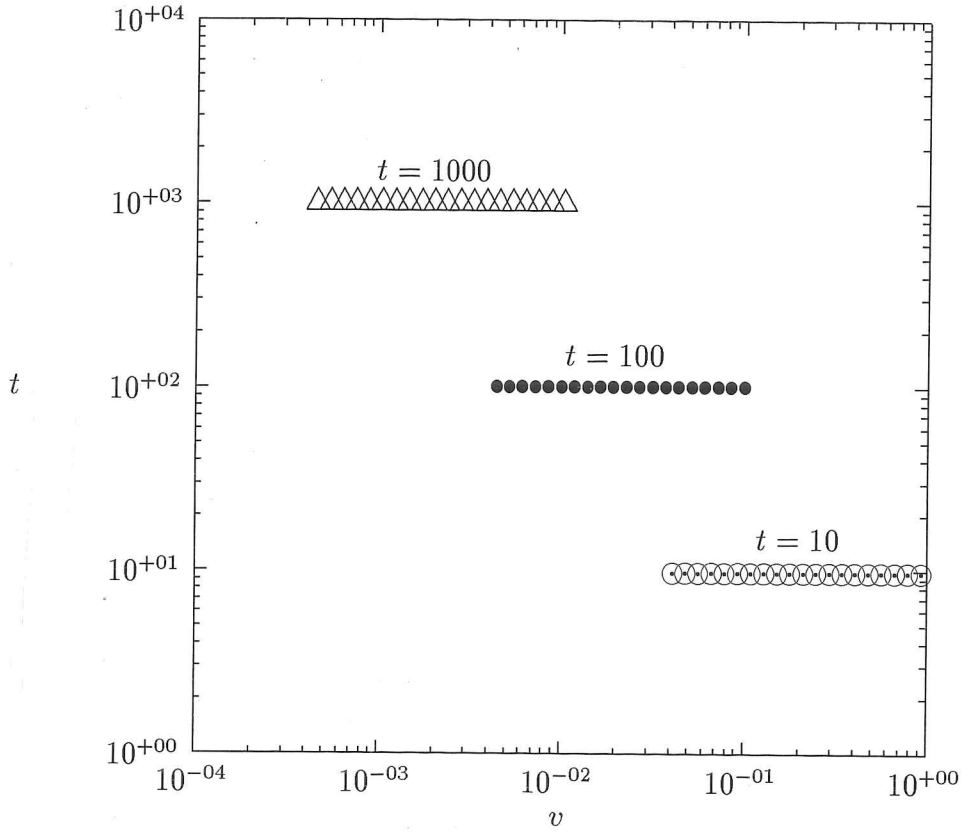

Figure 5.10: Locations of elements comprising the discretization over which the breakage problem was solved at times $t = 10, 100$ and 1000.

The manner in which the discretization of the domain changed with time is illustrated in figure 5.10. This figure shows the locations of the end-points of each element comprising a discretization at the times $t = 10, 100$ and 1000. At each time step the truncated domain has been stretched as described in subsection 5.12.3. The first node of each discretization is located at $v = 0$ however this is not shown since a logarithmic scale is used for the volume axis.

5.13.3 Case 3 : A Growth Problem

The dynamic PBE for growth is of the form :

$$\frac{\partial n(v,t)}{\partial t} + \frac{\partial}{\partial v}\left(G(v,t)n(v,t)\right) = 0 \tag{5.43}$$

If the initial distribution is selected to be :

$$n_{in}(v) = \frac{N_0}{v_0} v \exp(-v/v_0) \tag{5.44}$$

and the growth function is assumed to be independent of volume and time :

$$G(v,t) = G_0 \tag{5.45}$$

then the method of characteristics may be used to derive the analytical solution :

$$n(v,t) = \frac{N_0}{v_0}(v - G_0 t)\exp\left(-\frac{v - G_0 t}{v_0}\right) \qquad v \in (G_0 t, \infty] \tag{5.46}$$

Notice that the analytical solution is only considered within the domain $v \in (G_0 t, \infty]$. Outside of this region the solution assumes non-physical negative values.

Analytical expressions for the moments of this distribution may be obtained by multiplying the solution (5.46) by v^i (where i is the relevant index) and integrating over the domain $v \in (G_0 t, \infty]$. Using this procedure the first three moments may be evaluated as :

$$m_0(t) = N_0 v_0 \tag{5.47}$$

$$m_1(t) = N_0 v_0 (G_0 t + 2v_0) \tag{5.48}$$

$$m_2(t) = N_0 v_0 \left([G_0 t]^2 + 4 G_0 t v_0 + 6 v_0^2\right) \tag{5.49}$$

This problem was investigated for the following set of parameters :

$$N_0 = v_0 = 1 \qquad \text{and} \qquad G_0 = 0.01 \tag{5.50}$$

over the initial domain $v \in (0, 13.7]$ which was spanned by 20 elements with the first element of length 0.5 units.

Method	RKF	BS	Gear
CPU	51	$\otimes$	$\otimes$
steps	819	$\otimes$	$\otimes$
rL_2	0.52×10^{-2}	$\otimes$	$\otimes$

Table 5.3: CPU time, number of time steps required and the accuracy of the final solution when the Runge-Kutta-Fehlberg, the Bulirsch-Stoer and the Gear method were used to solve the dynamic growth problem to a time $t = 1000$.

Attempts were made to use the Runge-Kutta-Fehlberg (RKF), the Bulirsch-Stoer (BS) and the Gear methods to integrate the system of equations (5.26) that resulted from the collocation formulation of the growth PBE. Only the RKF method successfully completed the task. Negative nodal values were obtained in the tail region when using the other two methods, which prevented the domain from being successfully extrapolated. For the RKF method the CPU time (in seconds) required to integrate the problem to a final value of $t = 1000$ units, the number of time steps required and the rL_2 value of the solution at $t = 1000$ were recorded. These values are shown in table 5.3.

The finite element predictions of the density distribution are shown in figure 5.11 at three different times. The solid lines represent the analytical solution while the symbols ($\odot$) represent the FEM nodal values at the end-points of each element. These points were obtained using the Runge-Kutta-Fehlberg method to integrate the collocation formulation of equation (5.26). In each case the numerical predictions are in excellent agreement with the analytical solutions.

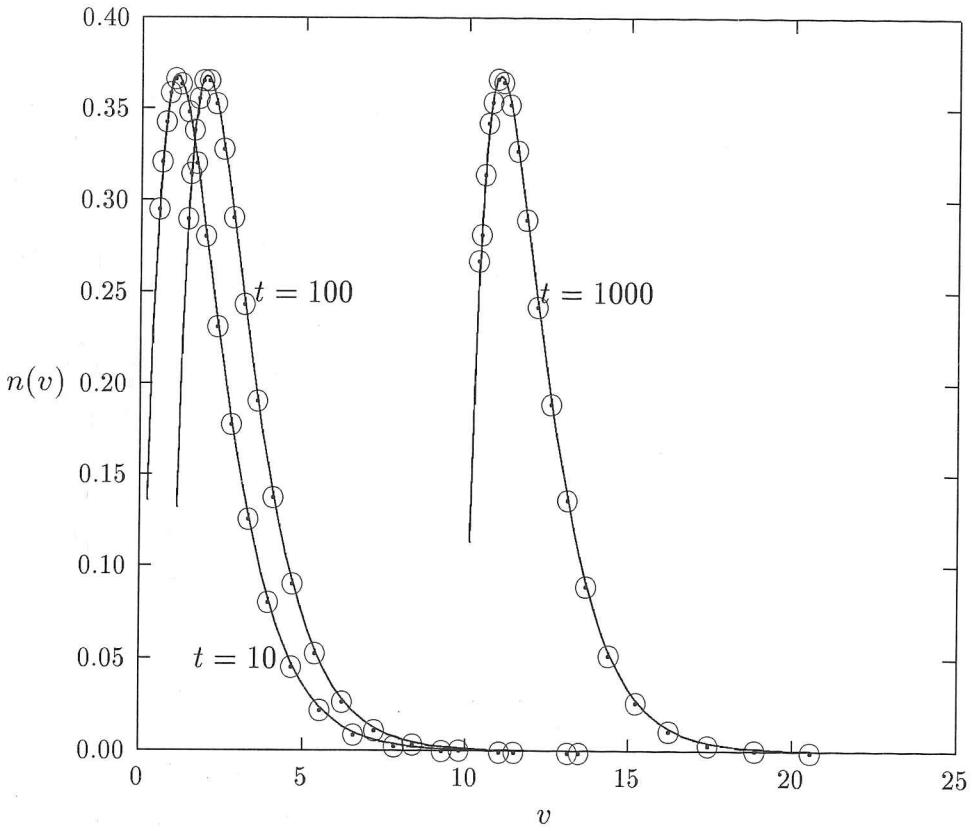

Figure 5.11: FEM predictions of the number density function $(n(v,t))$ for the dynamic growth problem where $G(v,t) = G_0 = 0.01$ $n_{in}(v) = \frac{N_0}{v_0} v \exp(-v/v_0)$ and $N_0 = v_0 = 1$ at times $t = 10, 100$ and 1000.

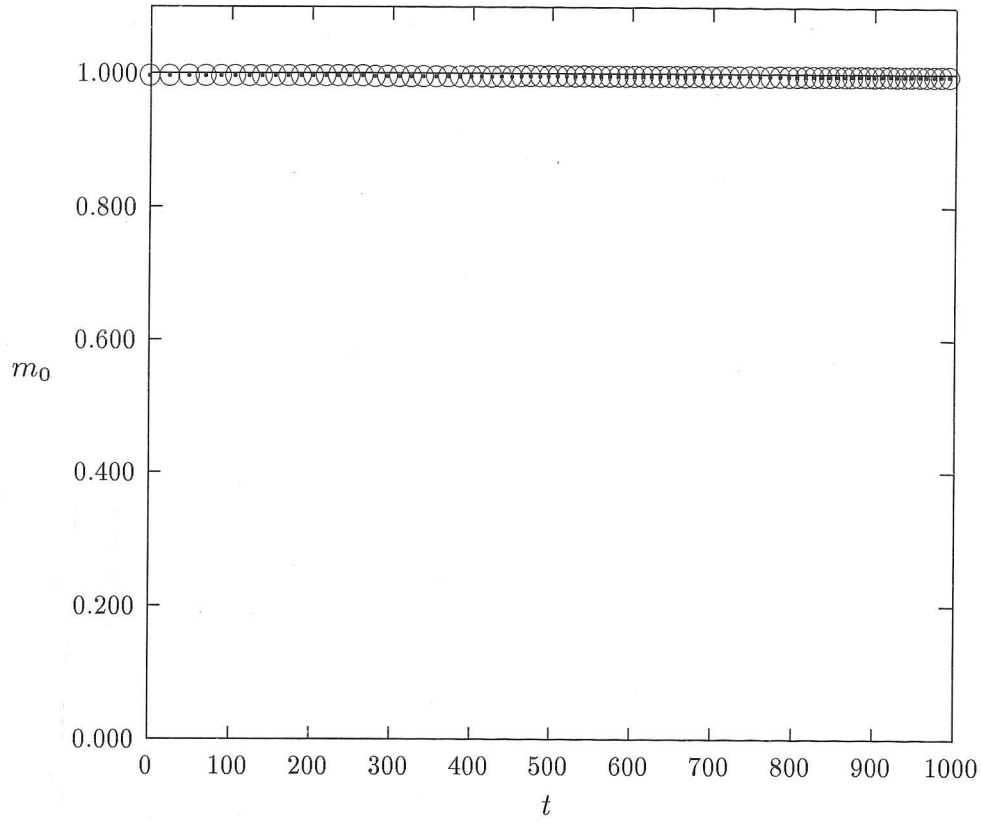

Figure 5.12: FEM predictions of the zeroth moment of the density distribution ($m_0(t)$) at every tenth time step for the dynamic growth problem.

Numerical approximations of the moments of the density distribution were obtained at each time step (t_s) by multiplying the finite element solution by v^i (where $i = 0, 1, 2$ for the zeroth, first and second moments) and integrating over the domain $v \in (G_0 t, v_{max}(t_s)]$. The finite element predictions of the moments are shown in figures 5.12, 5.13 and 5.14 as symbols ($\odot$) at every ten Runge-Kutta-Fehlberg time steps while the analytical expressions of the moments (5.47), (5.48) and (5.49) are plotted as solid lines. In each case the symbols representing the numerical values of the moments are barely discernable from the lines corresponding to the analytical expressions of the moments. The percentage errors in each of these moments at time $t = 1000$ was recorded in table 5.5 at the end of this section.

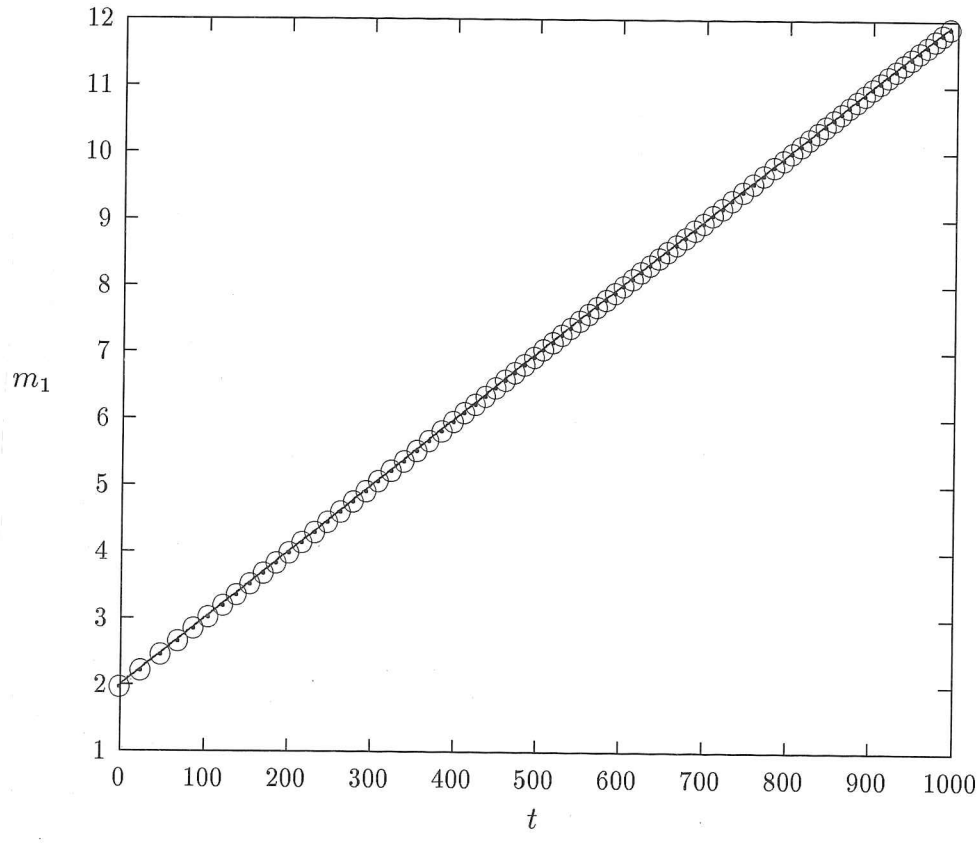

Figure 5.13: FEM predictions of the first moment of the density distribution ($m_1(t)$) at every tenth time step for the dynamic growth problem.

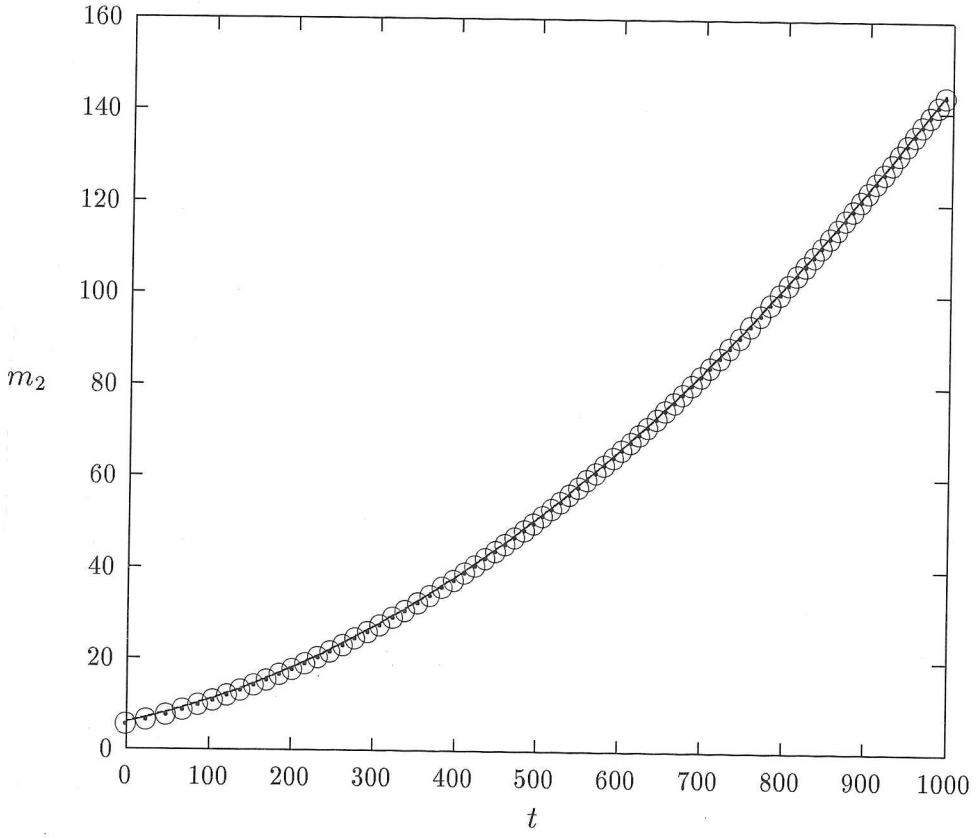

Figure 5.14: FEM predictions of the second moment of the density distribution ($m_2(t)$) at every tenth time step for the dynamic growth problem.

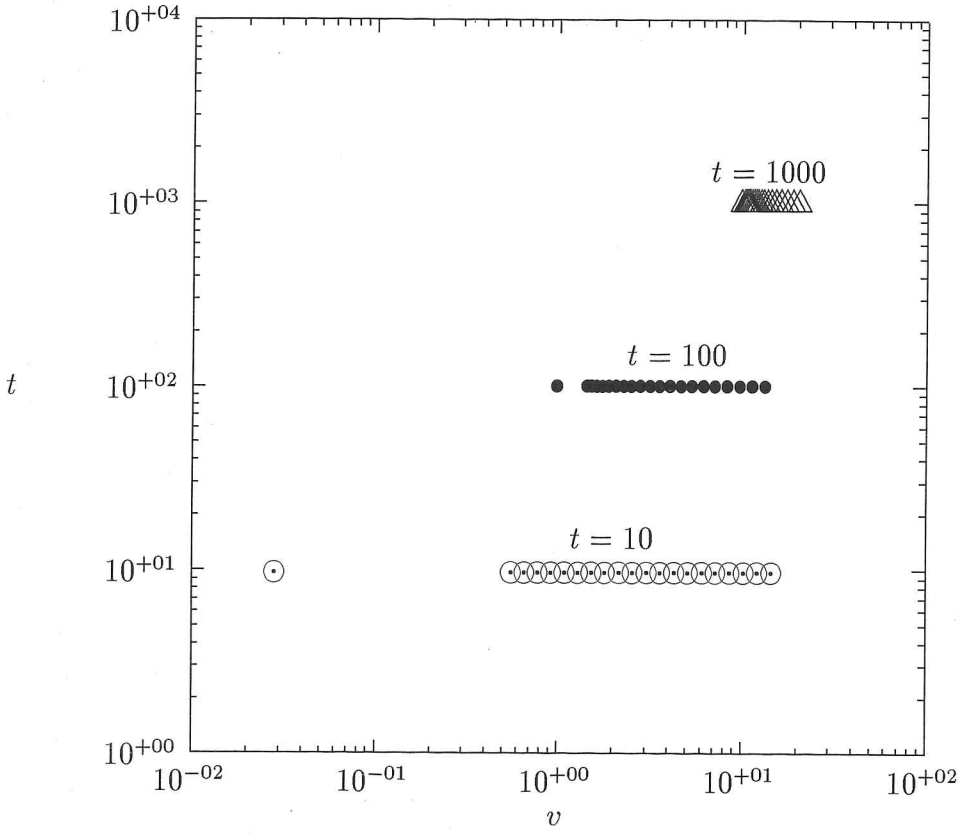

Figure 5.15: Locations of elements comprising the discretization over which the growth problem was solved at times $t = 10, 100$ and 1000.

The manner in which the truncated domain was adjusted is illustrated in figure 5.15. This figure shows the locations of the end-points of each element comprising a discretization at the times $t = 10, 100$ and 1000. In this problem the domain was stretched (as in the previous two case studies) and then translated so that the first node corresponded to the volume coordinate v_s such that $n(v_s) = 0$.

5.13.4 Case 4 : A Problem of Combined Aggregation and Growth

The dynamic PBE for aggregation and growth is of the form :

$$\frac{\partial n(v,t)}{\partial t} + \frac{\partial\big(G(v,t)n(v,t)\big)}{\partial v} = \frac{1}{2}\int_0^v \beta(v-w,w,t)n(v-w,t)n(w,t)dw - n(v,t)\int_0^\infty \beta(v,w,t)n(w,t)dw \tag{5.51}$$

If the initial distribution is selected to be :

$$n_{in}(v) = \frac{N_0}{v_0}\exp(-v/v_0) \tag{5.52}$$

and the growth function and aggregation kernel are assumed to be time independent and of the following forms :

$$G(v,t) = G_0 v \qquad \text{and} \qquad \beta(v,w,t) = \beta_0 \tag{5.53}$$

then the analytical solution as derived by Ramabhadran, Peterson and Seinfeld (1976) is :

$$n(v,t) = \frac{4N_0}{v_0(N_0\beta_0 t+2)^2}\exp\left(\frac{-2v\exp(-G_0 t)}{v_0(N_0\beta_0 t+2)} - G_0 t\right) \tag{5.54}$$

If the above solution is multiplied by v^i (where i is the relevant index) and integrated from zero to infinity then the analytical expressions for the first three moments may be deduced as :

$$m_0(t) = \frac{2N_0}{N_0\beta_0 t+2} \tag{5.55}$$

$$m_1(t) = N_0 v_0 \exp(G_0 t) \tag{5.56}$$

$$m_2(t) = N_0 v_0^2(N_0\beta_0 t+2)\exp(2G_0 t) \tag{5.57}$$

This problem was investigated for the following set of parameters :

$$N_0 = v_0 = 1, \qquad \beta_0 = 0.2 \qquad \text{and} \qquad G_0 = 2.5 \times 10^{-3} \tag{5.58}$$

over the initial domain $v \in (0, 11.3]$ which was spanned by 40 elements with the first element of length 0.75 units.

Method	RKF	BS	Gear
CPU	250	1156	914
steps	201	110	19
rL_2	0.73×10^{-2}	1.01×10^{-2}	2.38×10^{-2}

Table 5.4: CPU time, the number of time steps required and the accuracy of the final solution when the Runge-Kutta-Fehlberg, the Bulirsch-Stoer and the Gear method were used to solve the dynamic aggregation and growth problem to a time $t = 1000$.

The Runge-Kutta-Fehlberg (RKF), the Bulirsch-Stoer (BS) and the Gear method were each used to integrate the system of equations (5.26) that resulted from the collocation formulation of the combined aggregation and growth PBE. For each method the CPU time (in seconds) required to integrate the problem to a final value of $t = 1000$ units, the number of time steps required and the rL_2 value of the solution at $t = 1000$, were recorded. These values are shown in table 5.4. Once again the Runge-Kutta-Fehlberg method proved to be the fastest and most accurate means of solving the problem despite requiring significantly more time steps.

The finite element predictions of the density distribution are shown in figure 5.16 at three different times. The solid lines represent the analytical solution while the symbols ($\odot$) represent the FEM nodal values at the end-points of each element. These points were obtained using the Runge-Kutta-Fehlberg method to integrate the collocation formulation of equation (5.26). As in all the investigated cases the numerical predictions are in excellent agreement with the analytical solutions.

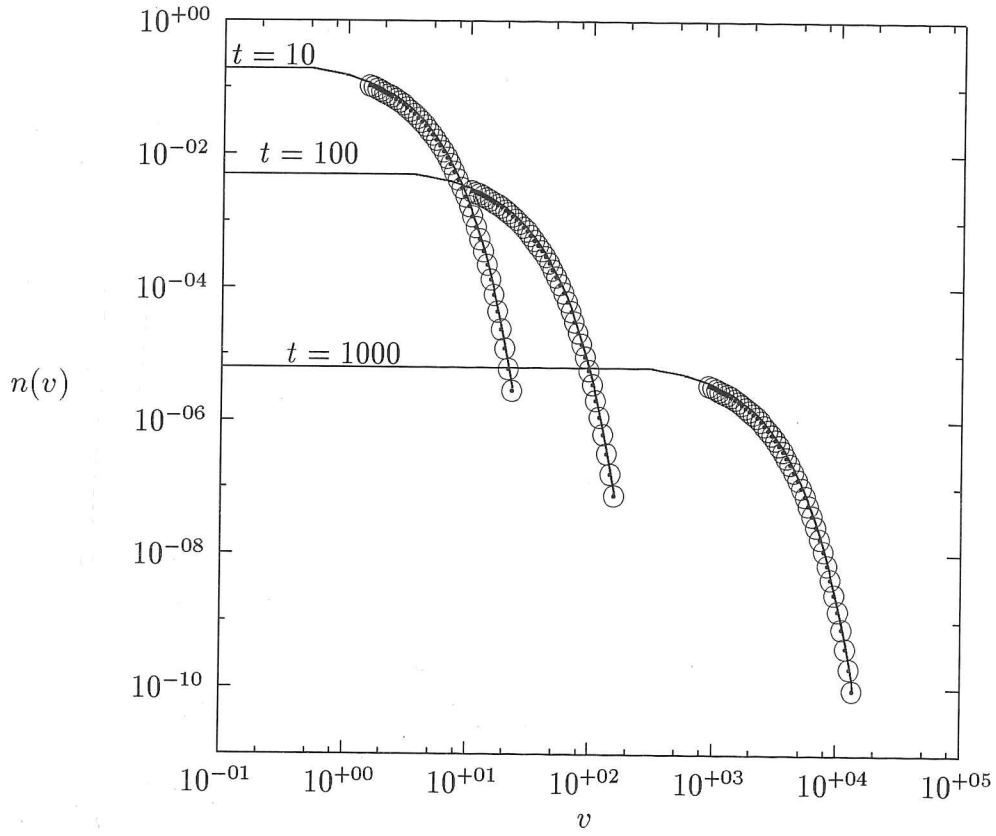

Figure 5.16: FEM predictions of the number density function ($n(v,t)$) for the dynamic aggregation and growth problem where $G(v,t) = G_0 v$, $\beta(v,w,t) = \beta_0$, $n_{in}(v) = \frac{N_0}{v_0}\exp(-v/v_0)$, $N_0 = v_0 = 1$, $\beta_0 = 0.2$ and $G_0 = 2.5 \times 10^{-3}$ at times $t = 10, 100$ and 1000.

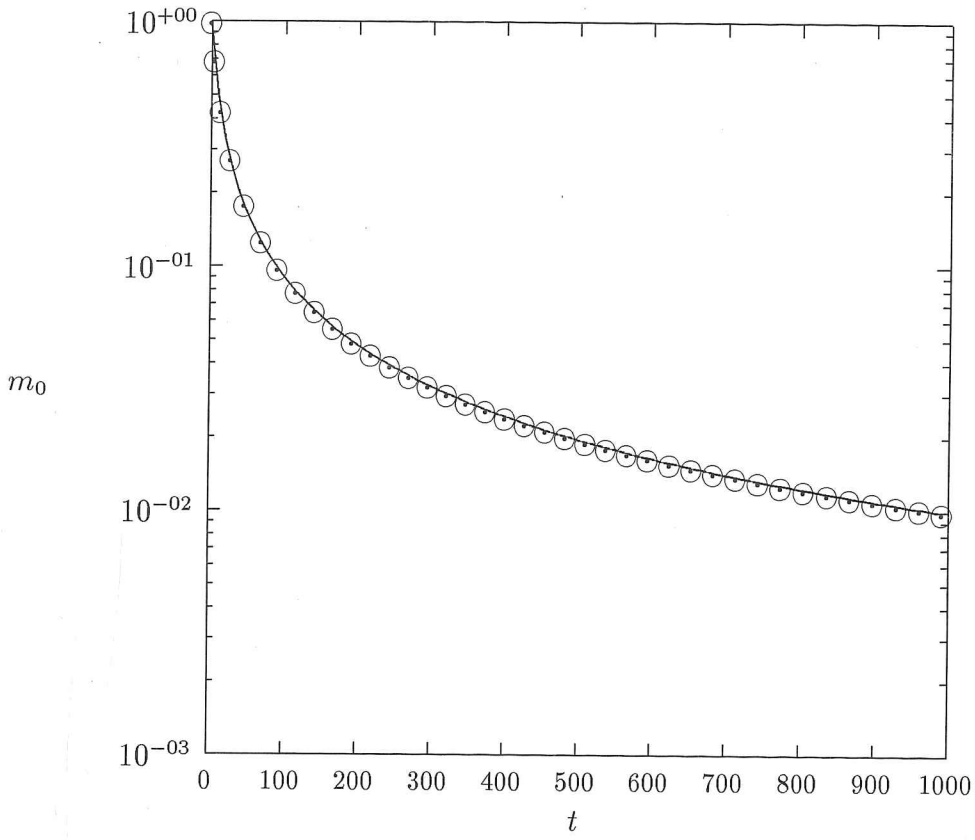

Figure 5.17: FEM predictions of the zeroth moment of the density distribution ($m_0(t)$) at every fifth time step for the dynamic aggregation and growth problem.

Numerical approximations of the moments of the density distribution were obtained at each time step (t_s) by multiplying the finite element solution by v^i (where $i = 0, 1, 2$ for the zeroth, first and second moments) and integrating over the domain $v \in (0, v_{max}(t_s)]$. The finite element predictions of the moments are shown in figures 5.17, 5.18 and 5.19 as symbols ($\odot$) at every fifth Runge-Kutta-Fehlberg time step while the analytical expressions of the moments (5.55), (5.56) and (5.57) are plotted as solid lines. In all cases the numerical predictions of the moments are barely discernible from the lines representing the analytical solutions. The percentage errors in each of these moments at time $t = 1000$ units are recorded in table 5.5 at the end of this section.

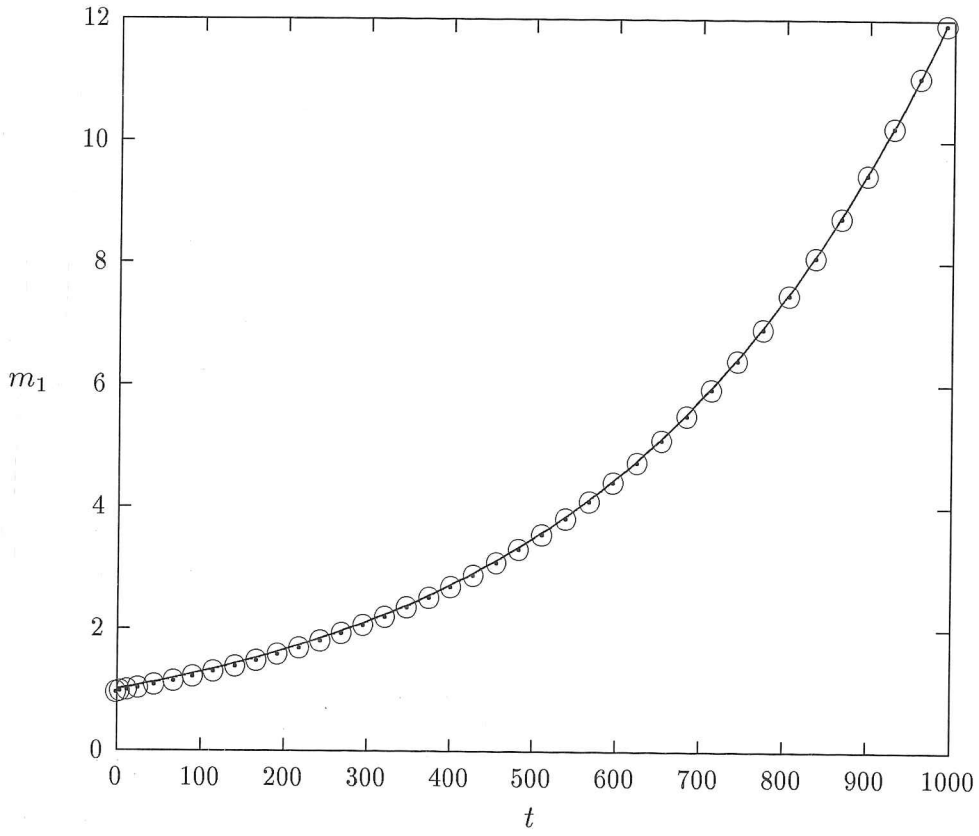

Figure 5.18: FEM predictions of the first moment of the density distribution ($m_1(t)$) at every fifth time step for the dynamic aggregation and growth problem.

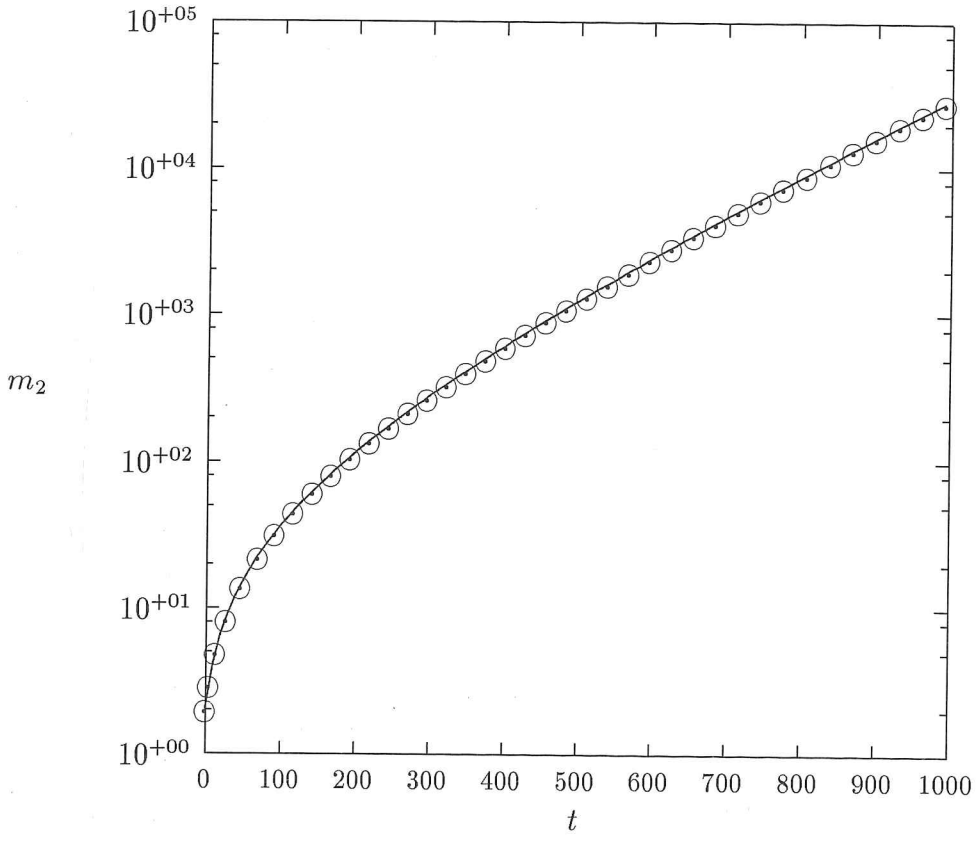

Figure 5.19: FEM predictions of the second moment of the density distribution ($m_2(t)$) at every fifth time step for the dynamic aggregation and growth problem.

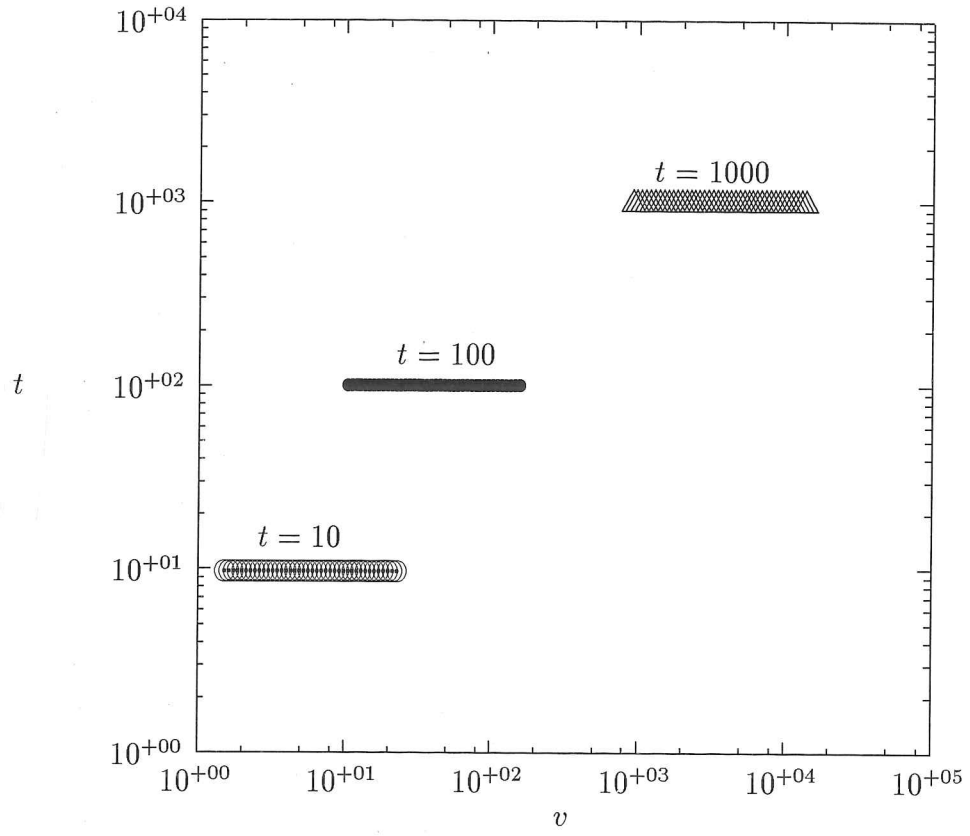

Figure 5.20: Locations of elements comprising the discretization over which the aggregation and growth problem was solved at times $t = 10, 100$ and 1000.

The manner in which the discretization of the domain changed with time is illustrated in figure 5.20. This figure shows the locations of the end-points of each element comprising a discretization at the times $t = 10, 100$ and 1000. In each case the lower limit of the first element is located at $v = 0$ however this is not shown since a logarithmic scale is used for the volume axis.

case	PE(m_0)	PE(m_1)	PE(m_2)
case 1	0.005 %	0.010 %	0.060 %
case 2	0.020 %	0.015 %	0.175 %
case 3	0.028 %	0.042 %	0.075 %
case 4	0.058 %	0.119 %	0.152 %

Table 5.5: Percentage errors in the zeroth, first and second moments at time $t = 1000$ units for each of the case studies.

In table 5.5 it can be seen that the finite element method proved capable of predicting the moments of the solutions to a range of problems to very high accuracy (better than 0.2% in all cases).

5.14 Discussion

5.14.1 Performance of the Time Integration Methods

In tables 5.1, 5.2, 5.3 and 5.4 it can be seen that the RKF method proved to be the fastest, most accurate and robust means of integrating the system of equations (5.26). When using the Gear method the total number of time steps required to solve the problem was drastically reduced (by as much as a factor of 15 in the breakage problem) however this gain was significantly out-weighed by the additional cost of numerically assembling and then manipulating a completely dense Jacobian matrix. In any case taking many small, inexpensive steps is preferable since the extrapolation procedure described in subsection 5.12.2 is performed over smaller intervals.

5.14.2 Accuracy and Computational Cost of the Finite Element Predictions

In all cases where the RKF method was used to integrate the system of equations (5.26) rL_2 values less than 1×10^{-2} were achieved (see tables 5.1, 5.2, 5.3 and 5.4). This is indicative of numerical solutions that are on average within 1% of their respective analytical solutions across the entire domain. In figures 5.1, 5.6, 5.11 and 5.16 the nodal values of the finite element solutions are barely discernable from the lines representing the analytical solutions. Likewise in figures 5.2-5.4, 5.7-5.9, 5.12-5.14 and 5.17-5.19 the finite element predictions of the moments of the density distributions are in excellent agreement with the analytical values. In table 5.5 it can be seen that at time $t = 1000$ units (where the greatest errors are likely to have accumulated) predictions of the moments were better than 0.200% in all the investigated cases.

It should be noted that solutions were obtained reasonably cheaply in cases 1 and 2 despite large changes in the initial distribution. In the aggregation problem a 500-fold reduction in the number of particles was observed. Despite these large changes an accurate solution was obtained using just 22 seconds of CPU time on a 75 MHz pentium processor. In the breakage problem a 1000-fold increase in the number of particles was observed nevertheless an accurate solution was obtained in just 76 seconds.

Solving PBEs where comparatively modest amounts of growth were occurred proved much more expensive. In case 3 a 10-fold increase in the total particulate volume required 51 seconds of CPU time while in case 4 a similar increase in volume occurred but 250 seconds of CPU time was required.

5.14.3 Scaling and Shifting of the Domain

In the simulations of the previous section the domains over which the PBEs were considered were adjusted so that criterion (2.47) was satisfied at each time step. In the aggregation problem for instance the domain was increased from 11.3 units to over 5600 units while in the breakage problem the domain was decreased from an initial length of 11.3 units to about 0.01 units. This scaling of the domain was performed to avoid the difficulties associated with solving the problem in regions where the solution assumed very small values and very steep profiles whilst negligibly increasing the accuracy of moment calculations. For instance if the same truncated domain $v \in (0, 11.3]$ was used for the entire breakage simulation we would be required to solve for nodal values of magnitude $n_j \approx 3.7 \times 10^{-4905}$ in the tail region as $t \rightarrow 1000$. Even the most robust ODE integrators would encounter difficulties with such a problem.

Likewise with the growth problem the lower limit of the domain of consideration was shifted to avoid solving for non-physical negative values.

In aggregation/growth dominant problems the domains were extended. This process required extrapolation of the solution and hence introduced additional errors to the numerical solution. These extrapolation errors can however be assumed to be quite small since such high quality solutions were obtained in all cases.

5.14.4 Comparison with Other Methods

In the majority of cases it is not possible to compare the predictions of the finite element method described in this chapter with other methods. In the articles reviewed in section 1.7 simulations have been performed using a large range of growth functions, breakage kernels etc. and carried out to different extents of aggregation, breakage, growth etc. In most cases no numerical values of errors are quoted. Instead results are presented graphically on scaled axes.

Two papers do however exist where the simulations performed were so similar to case 1 that comparisons should be made. Kumar and Ramkrishna (1996b) use their moving pivot method to solve an aggregation equation identical to case 1. This simulation is performed until a 250-fold reduction in total particle number is achieved ($N(t)/N(0) = 4 \times 10^{-3}$ in their notation). They report a 16% under-prediction in the zeroth moment (CON$(0,t)$ in their notation). This compares quite poorly with the prediction of the zeroth moment made by the finite element method (a 0.005 % over-prediction) when a 500-fold reduction in total particle number was achieved.

Litster, Smit and Hounslow (1995) investigated this same aggregation problem. In their simulation a 50-fold reduction in total particle number was achieved and they reported predictions of the second moment that were within 1 % of the analytical value. This simulation required 1077 units of CPU time on a SUN SLC Workstation. Again these figures compare quite poorly with those of the finite element method where the second moment was predicted to within 0.060 % of the analytical value for a much larger extent of aggregation (a 500-fold reduction in total particle number) using just 22 seconds of CPU time on a Pentium 75 MHz personal computer.

5.15 Chapter Conclusions

The following conclusions can be made with regard to the methods developed and simulations performed within this chapter :

- A finite element method has been developed to solve the dynamic population balance equation for the full range of particulate mechanisms : aggregation, breakage, growth and nucleation.
- The method has been derived over a general discretization in both time and space. Hence this method will be amenable to automatic refinement procedures should they be developed at a later stage.

- Of all the investigated time integration procedures the Runge-Kutta-Fehlberg proved to be the most accurate, robust and efficient method of integrating the system of ODEs that resulted from the collocation formulation of the dynamic PBE.

- In all the investigated problems the finite element method proved capable of predicting density distributions that were on average within 1 % of the analytical solution over the entire domain. Finite element predictions of the moments of these distributions were within 0.175 % of the analytical values.

- The above mentioned results were achieved over heuristically discretized domains. The space domain was geometrically partitioned while the size of the time steps was dictated by the local truncation error. Further increases in accuracy are anticipated upon incorporation of a more rigourous (error based) means of automatic discretization.

- Much of the success of the finite element method can be attributed to careful consideration of domain truncation points which are adjusted at each time step. This procedure prevents the system from becoming unnecessarily stiff whilst simultaneously ensuring that the finite domain errors are negligibly small.

- In all cases where comparisons could be made the finite element method was capable of predictions that were several orders of magnitude better than existing methods.

Chapter 6

Conclusions and Suggestions for Further Work

6.1 Conclusions : Or, what has actually been achieved

In the introductory chapter of this thesis the ultimate goal of modelling particulate processes was identified as one of developing a means of predicting the distributions of particles for any given set of process variables and any combination of particulate mechanisms. The work contained within this thesis represents a systematic advancement towards this ultimate goal for the following reasons :

- Methods capable of exceptionally high accuracy have been developed for solving both steady state and dynamic population balance equations for the density distribution and moments. Development of this capability is an essential stage in the modelling procedure since all parameterization techniques require an accurate solver.
- The developed methods are capable of solving the steady state and dynamic PBEs for any combination of particulate mechanisms (aggregation, breakage,

growth and nucleation) and for any choice of parameterizing functions (aggregation kernels, breakage kernels, growth functions etc) hence these methods are expected to be applicable to a large range of particulate problems.

- The developed methods have proved themselves more robust and computationally efficient than any existing methods. This factor is anticipated to become of increasing importance when modelling considerations are extended to higher coordinate systems (as will be discussed in the future applications section) or when the method is implemented in on-line control schemes.

- This body of work provides a full appreciation of the importance of precise domain specification. The proposed methods are only capable of solving the problem that is specified by the user. If the domain is poorly specified (truncated at an inappropriate point or inappropriately discretized) then the solution is guaranteed to be either of very poor quality or to have been excessively expensive to obtain. Problems associated with poorly specified domains are especially prevalent in PBE modelling due to the large variations in the size of sub-domains that comprise an "optimal" domain. It has been demonstrated in the simulations of chapters 2, 3 and 4 that when using a heuristic method of discretization it is very easy to under-specify the domain (use a discretization that is too sparse in some region) and obtain a solution that is inaccurate by several orders of magnitude. Over-specification of the domain (use of a discretization that is excessively fine in some regions) in many cases is not a viable alternative. In chapter 4 many geometric discretizations were over-specified in an attempt to achieve high accuracy. This over-specification rapidly resulted in problems that were too large to be solved using existing computer resources without ever achieving the required level of accuracy.

- This body of work is the first to rigorously address domain issues.

  In chapter 2 a criterion was proposed (and proved effective in the subsequent simulations of chapters 2, 3, 4 and 5) to maintain finite domain errors at negligibly small magnitudes while still permitting the moments of the solution to

be computed accurately. In the dynamic simulations of chapter 5, much of the robustness of the method can be directly attributed to the implementation of this truncation algorithm. By adjusting the truncation point at each time step the problem was prevented from ever becoming too stiff or ill-conditioned.

Chapter 4 was dedicated to the construction of a rigorous new approach to the important issue of domain discretization. This approach had the desirable characteristic of being automatic. In the simulations of chapter 4 it was demonstrated that the automatic discretization algorithm could be used to generate solutions with accuracies that could not be obtained using existing heuristic discretization methods.

- An error estimate was successfully derived in chapter 3. With this estimate it is possible to assess quantitatively the quality of an obtained numerical solution without any knowledge of the actual analytical solution. Much of the above-mentioned work on automatic discretization could not have eventuated without the developments of chapter 3 on error estimates. This work is the first to successfully relate the error in a numerical solution of the steady state PBE to the length of each element. It lies at the heart of the automatic discretization procedure and makes it possible for this procedure to generate numerical solutions with a pre-specified accuracy.

6.2 Recommendations for Further Work

The work on solving the steady state population balance equation (1.8) is essentially complete. A user can now specify the parameters of a problem, a maximum acceptable error tolerance and use the finite element method (together with the error estimate and automatic refinement algorithm) to obtain a numerical solution that is guaranteed to be of a pre-specified quality.

It is expected that future work on steady state particulate systems will progress in the following directions :

- *Inverse Problems* : In inverse problems the feed and product distributions are either known or can be measured. It is the parameterizing functions of the PBE that must be determined. If the forms of these functions are assumed then analytical moment relations may be used to determine the constants of each parameterizing function. The steady state solver can then be used to discriminate between the most likely parameterizing functions.

- *Higher Coordinate Systems* : The steady state PBE (1.8) is of the simplest possible form. Many of the assumptions inherent in its derivation limit its application to very ideal systems. It is expected that future work in this field will endeavour to incorporate more realistic features into population balance models. For instance modifications must be made to equation (1.8) if it is to simulate systems where aggregation, breakage or growth rates that are dependent upon temperature, concentration or shear stress profiles or if it is to model systems that are not well mixed.

The methods for solving the dynamic population balance equation were not developed to as advanced stage as the steady state methods. Further advances are needed in dynamic error estimation and automatic refinement techniques before the dynamic finite element method can be used with the same confidence as the steady state solver.

Once dynamic error estimates and automatic refinement methods are developed the dynamic inverse problem can be addressed and if necessary the PBE (1.7) can be modified to investigate less ideal systems.

Appendix : Analytical Solution to the Steady State Breakage Equation

Using the breakage function and specific rate of breakage selected in section 4.7.3 the PBE for breakage alone becomes :

$$\frac{n(v) - n_{in}(v)}{\tau} = 2\int_v^\infty n(w)dw - vn(v) \tag{6.1}$$

where $n_{in}(v)$ is the exponential feed distribution $\frac{N_0}{v_0}\exp(-v/v_0)$.

If the above PBE is multiplied by τ and differentiated with respect to v, the following first order ODE may be obtained :

$$\frac{dn(v)}{dv} + \frac{3\tau}{1+\tau v}n(v) = \frac{1}{1+\tau v}\frac{dn_{in}(v)}{dv} \tag{6.2}$$

By considering the breakage PBE (6.1) at the point $v = 0$ we obtain the appropriate boundary condition :

$$\begin{aligned} n_0 &= 2\tau m_0 + n_{in}(0) \\ &= 2\tau m_0 + \frac{N_0}{v_0} \end{aligned} \tag{6.3}$$

An expression for the zeroth moment is obtained by integrating (6.1) from zero to infinity :

$$\begin{aligned} m_0 &= \tau m_1 + m_0^{in} \\ &= \tau m_1^{in} + m_0^{in} \end{aligned} \tag{6.4}$$

The second equality holds true since volume is conserved in a breakage only problem.

Expressions for m_0^{in} and m_1^{in} may be readily obtained by integration of the exponential feed distribution :

$$\begin{aligned} m_0^{in} &= \int_0^\infty n_{in}(v)dv \\ &= \int_0^\infty \frac{N_0}{v_0} \exp(-v/v_0)dv \\ &= N_0 \end{aligned} \tag{6.5}$$

$$\begin{aligned} m_1^{in} &= \int_0^\infty v\, n_{in}(v)dv \\ &= \int_0^\infty \frac{N_0}{v_0} v \exp(-v/v_0)dv \\ &= N_0 v_0 \end{aligned} \tag{6.6}$$

Expressions (6.3, 6.4, 6.5 and 6.6) are combined to obtain the boundary condition in terms of the problem parameters :

$$n_0 = 2\tau N_0(\tau v_0 + 1) + \frac{N_0}{v_0} \tag{6.7}$$

The integrating factor $I(v)$ is evaluated as :

$$I(v) = \exp\left(\int_0^v \frac{3\tau}{1+\tau w} dw\right) = (1+\tau v)^3 \tag{6.8}$$

The solution to the ODE (6.2) with the boundary condition (6.7) is :

$$n(v) = \frac{n_0}{I(v)} + \frac{1}{I(v)} \int_0^v \frac{I(t)}{1+\tau t} \frac{dn_{in}(t)}{dv} dt \tag{6.9}$$

Integration and algebraic manipulation of this expression yields the solution to the breakage PBE (6.1) :

$$n(v) = \frac{N_0[(1+\tau v)^2 + 2\tau v_0(1+\tau[v_0+v])]}{v_0(1+\tau v)^3 \exp\left(\frac{v}{v_0}\right)} \tag{6.10}$$

Notation

b	=	birth rate
b^a	=	birth rate due to aggregation
b^b	=	birth rate due to breakage
d	=	death rate
d^a	=	death rate due to aggregation
d^b	=	death rate due to breakage
$PE(m_i)$	=	percentage error in the prediction of the i^{th} moment
G	=	growth function
G_0	=	growth constant
h_e	=	length of the e^{th} element
I_0	=	modified Bessel function of the first kind of zeroth order
I_1	=	modified Bessel function of the first kind of first order
I_{agg}	=	index of aggregation
I_{bre}	=	index of breakage
$m(i)_0$	=	number of particles in the i^{th} partition
m_i	=	i^{it} moment of the density distribution
m_i^{dte}	=	error in the i^{th} moment due to domain truncation
N	=	number of elements used to span the domain
N_0	=	number of nuclei present in the exponential feed distribution

n = number density distribution of the product stream

n_0 = number of nuclei

n_{in} = number density distribution of the feed stream or the initial distribution

n_h = finite element solution of the PBE

n_j^e = j^{th} nodal value of element e

N_{eq} = number of equations comprising the system

N_{it} = number of iterations required for the system to converge

S = specific rate of breakage

t = time

v = particle volume

v_1 = length of the first element in the GP of the FEM

v_a^e = volume co-ordinate of the lower limit of element e

v_b^e = volume co-ordinate of the upper limit of element e

v_{max} = upper limit of the truncated domain

v_{max}^0 = initial estimate of the upper limit of the truncated domain

v_0 = mean size of the nuclei in the exponential feed distribution

Greek symbols

τ = time constant of the CMSMPR crystallizer

β = aggregation kernel

β_0 = size independent aggregation kernel

Δ = maximum permissible REE value

ρ = breakage function

ϕ_j^e = the j^{th} Lagrange interpolation polynomial of the e^{th} element

ψ_i^e = the i^{th} weight function

Ω_e = domain of the e^{th} element

Subscripts and Superscripts

a	=	pertaining to the first node
b	=	pertaining to the last node
ana	=	pertaining to the analytical solution
e	=	pertaining to the e^{th} element
fea	=	pertaining to the finite element solution
g	=	element of a global matrix or vector
s	=	the s^{th} iteration or step

Acronyms

ARD	=	Automatically discretized domain
FEM	=	Finite element method
GRD	=	Geometrically discretized domain
CPU	=	Central processing unit
DPB	=	Discretized population balance
PBE	=	Population balance equation
REE	=	Relative error estimate
REN	=	Relative error norm
SSRE	=	Sum of square relative errors

Bibliography

[1] Adams, R.A., (1975), *Sobolev Spaces*, Academic Press.

[2] Bass, L., (1954), Zur theorie der mahlvorgänge, *Zeits. Angew Maths. Phys.*, **5**,283-292.

[3] Batterham, R.J.,' Hall, J.S. and Barton, G., (1981), Pelletizing Kinetics and Simulation of Full-Scale Balling Circuits, *Proceeding of the 3rd International Symposium on Agglomeration*, Nürnberg, W. Germany, A136.

[4] Bhatia, S.K. and Chakraborty, D., (1996), Modified MWR approach : Application to agglomerative precipitation, *A.I.Ch.E. Journal*, **38**, 6, 868-878.

[5] Chen, M.Q., Hwang, C. and Shih, Y.P., (1996), A wavelet-Galerkin method for solving population balance equations, *Computers and Chemical Engineering*, **20**, 2, 131-145.

[6] Ciarlet, P.G., (1978), *The finite element method for elliptic problems*, North-Holland, Amsterdam.

[7] Ciarlet, P.G., Schultz, M.H. and Varga, R.S., (1969), Numerical methods of higher order accuracy for nonlinear boundary value problems. V. Monotone operator theory, *Numer. Math.*, **13**, 51-77.

[8] Drake, R.L., (1972), A general mathematical survey of the coagulation equation, *Topics in Current Aerosol Research*, Part 2, **3**, Hidy, G.M. and Brock, J.R., Pergamon, New York.

[9] Erasmus, L.D., Eyre, D. and Everson, R.C., (1994), Numerical treatment of the population balance equation using a spline-Galerkin method, *Computers and Chemical Engineering*, **18**, 9, 775-783.

[10] Eyre, E., Wright, C.J. and Reuter, G., (1988), Spline-collocation with adaptive mesh grading for solving the stochastic collection equation, *Journal of Computational Physics*, **78**, 288-304.

[11] Filippov, A.F., (1961), On the distribution of the sizes of particles which undergo splitting, *Theory of probability and its applications*, **6**, 275-294.

[12] Gaudin, A.M. and Meloy, T.P., (1962), Model and comminution distribution equation for repeated fracture, *Trans. AIME*, **223**, 43-50.

[13] Gardner, R.P. and Austin, L.G., (1962), A chemical engineering treatment of batch grinding, Part I, *Symposium Serkleinern*, 217-231.

[14] Gear, C.W., (1971), *Numerical initial value problems in ordinary differential equations*, Prentice-Hall, NJ.

[15] Gelbard, F. and Seinfeld, J.H., (1978), Numerical solution of the dynamic equation for particulate systems, *Journal of Computational Physics*, **28**, 357-375.

[16] Gelbard, F., Tambour, Y. and Seinfeld, J.H., (1980), Sectional representations for simulating aerosol dynamics, *Journal of Colloid and Interface Science*, **76**, 2, 541-556.

[17] Gerald, C.F. and Wheatley, P.O., (1989), *Applied numerical analysis*, 4^{th} edition, Addison-Wesley.

[18] Hill, P.J. and Ng, K,M.,(1995), New discretization procedure for the breakage equation, *A.I.Ch.E. Journal*, **41**, 5, 1204-1216.

[19] Hill, P.J., (1996), Statistics of multiple particle breakage, *A.I.Ch.E. Journal*, **42**, 6, 1600-1611.

[20] Hounslow, M.J., (1990), A discretized population balance for continuous systems at steady state, *A.I.Ch.E. Journal*, **36**, 1, 106-116.

[21] Hounslow, M.J., (1997), Asymptotic particle size distributions for size independent aggregation and growth in a CST, *in progress.*

[22] Hounslow, M.J., Ryall, R.L. and Marshall, V.R., (1988), A discretized population balance for nucleation, growth and aggregation, *A.I.Ch.E. Journal*, **34**, 11, 1821-1832.

[23] Hulburt, H.M. and Katz, S., (1964), Some problems in particle technology : A statistical mechanical formulation., *Chemical Engineering Science*, **19**, 555-74.

[24] Hwang, C., and Shih, Y.P., (1982), Solution of population balance equations via block pulse function, *The Chemical Engineering Journal*, **25**, 39-45.

[25] Kaps, P. and Rentrop, P., (1979), *Numerische Mathematik*, **33**, 55-68.

[26] Kim, W.S. and Tarbell, J.M., (1990), Numerical technique for solving population balances in precipitation processes, *Chemical Engineering Communications*, **101**, 115-129.

[27] Kumar, S., and Ramkrishna, D., (1996a), On the solution of population balance equations by discretization—I. A fixed pivot technique, *Chemical Engineering Science*, **51**, 8, 1311-1322.

[28] Kumar, S., and Ramkrishna, D., (1996b), On the solution of population balance equations by discretization—II. A moving pivot technique, *Chemical Engineering Science*, **51**, 8, 1311-1322.

[29] Landgrebe, J.D., and Pratsinis, S.E., (1990), A discrete-sectional model for particulate production by gas-phase chemical reaction and aerosol coagulation in the free-molecular regime, *Journal of Colloid and Interface Science*, **139**, 1, 63-86.

[30] Liao, P.F. and Hulburt, H.M., (1976), Agglomeration processes in suspension crystallization. Presented at the Annual General Meeting of the AIChE, Chicago, December 1976.

[31] Litster, J.D., Smit., D.J. and Hounslow, M.J., (1995), Adjustable discretized population balance for growth and aggregation, *A.I.Ch.E. Journal*, **41**, 3, 1-13.

[32] Melkes, F., (1970), The finite element method for nonlinear problems, *Apl. Mat.*, **15**, 117-189.

[33] Melzak, Z.A., (1957), A scalar transport equation, *Trans. Am. Math. Soc.*, **85**, 547-60.

[34] Muhr, H., David, R., Villermaux, J. and Jezequel, P.H., (1996), Crystallization and precipitation engineering—VI. Solving population balance in the case of the precipitation of silver bromide crystals with high primary nucleation rates by using first order upwind differentiation, *Chemical Engineering Science*, **51**, 2, 309-319.

[35] Müller, H., (1928), Zur allgemeinen Theorie der raschen Koagulation, *Kolloidchemische Beihefte*, **27**, 223-50.

[36] Nicmanis, M., (1995), A finite element analysis of the steady state population balance equation, *Certificate of Post-Graduate Studies*, Report submitted to the Department of Chemical Engineering, University of Cambridge.

[37] Noor, M.A. and Whiteman, J.R., (1976), Error bounds for finite element solution of mildly nonlinear elliptic boundary value problems., *Numer. Math.*, **26**, 107-116.

[38] Pilinis, C., (1990), Derivation and numerical solution of the species mass distribution equations for multicomponent particulate systems, *Atmospheric Environment*, **24A**, 7, 1923-1928.

[39] Prasher, C.L., (1987), *Crushing and Grinding Process Handbook*, Wiley, New York.

[40] Press, W.H., Teukolsky, S.A., Vetterling, W.T. and Flannery, B.P., (1992), *Numerical Recipes in Fortran*, 2nd edition, Cambridge University Press.

[41] Randolph, A.D. and Larson, M.A., (1988), Theory of Particulate Process, 2nd ed., Academic Press, San Diego.

[42] Ramabhadran, T.E., Peterson, T.W. and Seinfeld, J.H., (1976), Dynamics of aerosol coagulation and condensation, *AIChE Journal*, **22**, 840-851.

[43] Reddy, J.N., (1993), An introduction to the finite element method, 2nd ed., McGraw Hill.

[44] Sampson, K.J. and Ramkrishna, D., (1984), A new solution to the Brownian coagulation equation through the use of root-shifted problem-specific polynomials, *Journal of Colloid Interface Science*, **103**, 1, 245-254.

[45] Sastry, K.V.S. and Gaschignard, P., (1981), Discretization procedure for the coalescence equation of particulate processes, *Ind. Eng. Chem. Fundam.*, **20**, 4, 355-361.

[46] Scott, W.T., (1968), Analytic studies of cloud droplet coalescence I, *J. Atmos. Sci.*, **25**, 54-65

[47] Singh, P.N. and Ramkrishna, D., (1975), Transient solution of the Brownian coagulation equation by problem-specific polynomials, *Journal of Colloid and Interface Science*, **53**, 2, 214-223.

[48] Singh, P.N. and Ramkrishna, D., (1977), Solution of population balance equations by MWR, *Computers and Chemical Engineering*, **1**, 23-31.

[49] Smit, D.J., Hounslow, M.J. and Paterson, W.R., (1993), Aggregation and gelation 1 : Analytical solutions for CST and batch operations, *Chemical Engineering Science*, **49**, 7, 1025-1035.

[50] Smoluchowski, M.V., (1916), Drei Vorträge über Diffusion, Brownsche Bewegung und Koagulation von Kolloidteilchen, *Physik. Z.*, **17**, 557-85.

[51] Smoluchowski, M.V., (1917), Versuch einer mathematischen Theorie der Koagulationskinetic kolloider Lösunger, *Z. Phys. Chem.*, **92**, 129-68.

[52] Steemson, M.L. and White, E.T., (1988), Numerical modelling of steady state continuous crystallization processes using piecewise cubic spline functions, *Computers and Chemical Engineering*, **12**, 1, 81-89.

[53] Stoer, J., and Bulirsch, R., (1980), *Introduction to Numerical Analysis*, Springer-Verlag, New York.

[54] Tsang, T.H. and Brock, J.R., (1983), Simulations of condensation and evaporation of condensation aerosol, *Aerosol Science and Technology*, **2**, 311-320.

[55] Viljoen, H.J., Eyre, D. and Wright, C.J., (1990), Solving dynamic equations for the collection and evaporation of an aerosol, *The Canadian Journal of Chemical Engineering*, **68**, 938-943.

[56] Wang, M.L. and Chang, R.Y., (1983), Legendre function approximations of ordinary differential equation and application to continuous crystallization process, *Chem. Eng. Commun.*, **22**, 115-125.

[57] Wang, M.L. and Chang, R.Y., (1987), Modelling of Crystallization systems via generalized orthogonal polynomials method, *Chem. Eng. Comm.*, **57**, 197-213.

[58] Warren, D.R., and Seinfeld, J.H., (1985), Simulation of aerosol size distribution evolution in systems with simultaneous nucleation, condensation and coagulation, *Aerosol Science and Technology*, **4**,31-43.

[59] Wu, J.J. and Flagan, R., (1988), A discrete-sectional solution to the aerosol dynamic equation, *Journal of Colloid and Interface Science*, **123**, 2, 339-352.

[60] Wynn, E.J.W., (1996), Improved accuracy and convergence of discretized population balance of Litster et al., *A.I.Ch.E. Journal*, **42**, 7, 2084-2086.

[61] Zeidler, E., (1995), *Applied functional analysis : applications to mathematical physics*, Springer-Verlag, New York.

[62] Zenisek, A., (1990), *Nonlinear elliptic and evolution problems and their finite element approximations*, Academic Press, London.

[63] Zienkiewicz, O.C. and Taylor, R.L., (1989), *The Finite Element Method*, vol. 1 : *Linear Problems*, 4th edition, McGraw Hill, New York.

[64] Ziff, R.M. and McGrady, E.D., (1985), The kinetics of cluster fragmentation and depolymerization, *J. Phys. A: Math. Gen.*, **18**, 3027-3037.